Computational Methods for Multilinear Forms in Pattern Recognition of Multiway Data

January 2017

Hayato Itoh

Graduate School of Advanced Integrated Science,
Chiba University

（千葉大学審査学位論文）

Computational Methods for Multilinear Forms in Pattern Recognition of Multiway Data

January 2017

Hayato Itoh

Graduate School of Advanced Integrated Science,
Chiba University

Acknowledgements

I would like to express gratitude to my many friends and colleagues for their helpful advice and discussion. In particular I would like to express my thanks to Professor Takashi Imaizumi at Chiba University, the chair of the examiners, who supported my research activity at Institute of Management and Information Technologies, Chiba University. Many thanks are also due to all the examiners, Professors Hiroki Suyari, Tadashi Yamaguchi, Yasuharu Den at Chiba University, and Professor Akihiro Sugimoto at National Institute of Infomatics, for their useful comments to improve this dissertation. Furthermore, in particular I would like to express my thanks to Professor Akihiro Sugimoto, who supported my research activity at National Institute of Infomatics. Moreover, I would like to express my thanks to Associate Professor Kazuhiko Kawamoto at Chiba Univeristy and Associate Professor Tomoya Sakai at Nagasaki Univeristy, the co-supervisors, for their useful comments to improve this dissertation.

I am also grateful to Professor Václav Hlaváč, Tomas Pajdla, Alexander Shekhovtsov, Professor Jiří Matas and Professor Mirko Navara, at Center for Machine Perception, Czech Technical University in Prague, for their helpful advice and hospitality during my stay in their institution. During my stay in their institution, I learned many things with friends in the institution through powerful support and creative atmosphere.

I am most indebted to my supervisor, Professor Atsushi Imiya at Chiba University, who initiated my interest in the study of multilinear pattern recognition based on functional analysis and multilinear algebra. This dissertation would not have been accomplished without his guidance.

This dissertation was supported by the "Computational Anatomy for Computer-Aided Diagnosis and Therapy: Frontiers of Medical Image Sciences" and "Multidisciplinary Computational Anatomy and Its Application to Highly Intelligent Diagnosis and Therapy" projects funded by a Grant-in-Aid for Scientific Research on Innovative Areas from MEXT, Japan, and by Grants-in-Aid for Scientific Research funded by the Japan Society for the Promotion of Science.

Contents

1 Introduction **30**
- 1.1 Background and Purpose . . . 30
- 1.2 Related Works . . . 33
- 1.3 Organization of the Dissertation . . . 37

2 Multilinear Forms ofPattern **39**
- 2.1 Multilinear Form . . . 39
  - 2.1.1 Preliminaries . . . 39
  - 2.1.2 First-Order Tensor . . . 42
  - 2.1.3 Second-Order Tensor . . . 42
  - 2.1.4 Third-Order Tensor . . . 44
  - 2.1.5 Nth-Order Tensor . . . 48
- 2.2 Principal Component Analysis . . . 51
  - 2.2.1 First-Order Tensor . . . 51
  - 2.2.2 Second-Order Tensor . . . 53
  - 2.2.3 Third-Order Tensor . . . 59
  - 2.2.4 Nth-Order Tensor . . . 62
- 2.3 Discrete Cosine Transform . . . 66
  - 2.3.1 One-Dimensional Discrete Cosine Transform . . . 66
  - 2.3.2 Two-Dimensional Discrete Cosine Transform . . . 68
  - 2.3.3 Three-Dimensional Discrete Cosine Transform . . . 68
  - 2.3.4 N-Dimensional Discrete Cosine Transform . . . 69
  - 2.3.5 Relation to Scale Space and Pyramid Transform . . . 70

3 Recognition of Bilinear Forms **75**
- 3.1 Dimension Reduction Methods for Image Pattern Recognition 75
- 3.2 Related Works . . . 78
- 3.3 Mathematical Preliminaries . . . 82
  - 3.3.1 Pyramid Transform . . . 82
  - 3.3.2 Random Projection . . . 85
  - 3.3.3 Two-Dimensional Random Projection . . . 88

3.4 Topology and Geometry in Pattern Recognition 89
3.5 Classification Methods 93
3.5.1 Subspace Method 93
3.5.2 Mutual Subspace Method 94
3.5.3 Constraint Mutual Subspace Method 96
3.5.4 Tensor Subspace Methods 97
3.6 Experiments 98
3.7 Summary 107

4 Recognition of Multilinear Forms 125
4.1 Recognition Methods for Multilinear Forms 125
4.2 Related Works 126
4.3 Tensor Subspace of Categories 128
4.4 Tensor Subspace Method 128
4.5 Tensor Subspace of Queries 129
4.6 Mutual Tensor Subspace Method 129
4.7 Geometry of Multilinear Subspace 130
4.8 Experiments 136
4.8.1 Gait Patterns 136
4.8.2 Volumetric Pattern 141
4.8.3 Spatio-Temporal Pattern 147
4.8.4 Geometry of Multilinear Subspace 152
4.9 Summary 179

5 Feature Extraction 182
5.1 Feature Extraction Methods 182
5.2 Related Works 184
5.3 Mathematical Preliminaries 186
5.3.1 Function Space 186
5.3.2 Directional Gradient and Structure Tensor 188
5.3.3 Aggregation Methods 191
5.3.4 Distribution of Directional Gradient 193
5.3.5 Distribution of Dominant Directional Gradient 194
5.3.6 Gradient-Based Discrimination 195
5.4 Feature Extraction and Discrimination 195
5.4.1 Local Directional Distribution Methods 195
5.4.2 Global Directional Distribution Method 198
5.4.3 HOG-based Discrimination 199
5.5 Experiments 202
5.6 Discussion 206

5.7 Summary 213

6 Estimation of Geometrical Transform 215
6.1 Manifold Learning for Image Registration 215
6.2 Related Works 218
6.3 Global Image Registration 219
6.4 Local Eigenspace 220
6.4.1 Two-Dimensional Image 220
6.4.2 Three-Dimensional Image 221
6.5 Affine Transformation 222
6.5.1 Two-Dimensional Image 222
6.5.2 Three-Dimensional Image 223
6.6 Neighbours of Template Image 224
6.7 Manifold Generation by Random Projection 224
6.8 Local Linear Method 226
6.8.1 Two-Dimensional Image 226
6.8.2 Three-Dimensional Image 228
6.9 Experiments 231
6.9.1 Two-Dimensional Image 231
6.9.2 Three-Dimensional Image 235
6.10 Summary 239

7 Conclusions 242

Bibliography 243

Publications 261

List of Figures

1.1 Sampling, vectors and tensors. The sampled value $f(\Delta \boldsymbol{z}), \boldsymbol{z} \in \mathbb{Z}^n$ of a function $f(\boldsymbol{x}), \boldsymbol{x} \in \mathbb{R}^n$ yields an array $f_{\boldsymbol{z}}, \boldsymbol{z} \in \mathbb{Z}^{m \times m \times \cdots \times m}$. This array $f_{\boldsymbol{z}}$ is expressed as a tensor $\mathcal{F}$ to preserve multilinearity of $f(\boldsymbol{x})$. Interpolation procedure reconstruct $f(\boldsymbol{x})$ from $Sf(\boldsymbol{x})$ through $\mathcal{F}$. The vector $\boldsymbol{f}$ whose elements are sample values of $f(\boldsymbol{z})$ is constructed from $\mathcal{F}$ by vectorisation operator vec to the tensor $\mathcal{F}$. 32

1.2 The relations among chapters in this dissertation. 38

2.1 Examples of tensors. (a) a first-order tensor, that is a vector. (b) a second-order tensor, that is a matrix. (c) a third-order tensor. (d) a fourth-order tensor. (3) an Nth-order tensor, which is a sequence of $(N-1)$th-order tensors. 41

2.2 1- and 2-mode unfoldings of a second-order tensor $\mathcal{X} \in \mathbb{R}^{6 \times 8}$. 45

2.3 Vectorising of a second-order tensor. By connecting unfolded 1-mode vectors, we have a long vector. 45

2.4 Tensor-tensor projection of a second-order tensor $\mathcal{X} \in \mathbb{R}^{6 \times 8}$ to a lower-dimensional tensor $\mathcal{Y} \in \mathbb{R}^3$. 46

2.5 Unfoldings of a third-order tensor showing 1-, 2- and 3-mode unfoldings of the third-order tensor $\mathcal{X} \in \mathbb{R}^{4 \times 5 \times 3}$. 48

2.6 Vectorising of a third-order tensor. By connecting unfolded 1-mode vectors, we have a long vector. 49

2.7 Tensor-tensor projection of a third-order tensor $\mathcal{X} \in \mathbb{R}^{4 \times 5 \times 3}$ to a lower-dimensional tensor $\mathcal{Y} \in \mathbb{R}^{3 \times 2 \times 1}$. 50

2.8 Image representation and dimension reduction. In the origin of the flow, $e_{ij}, e_{i'j'}, e_{i''j''}$ are the basis representing each pixel of an image f. After vectorisation of the image, $e_k, e_{k'}, e_{k''}$ are the standard basis for the Euclidean space for the vectorised image $\mathrm{vec}\, f$. After 1-mode unfolding, $u_i^{(1)}, u_{i'}^{(1)}$ and $u_{i''}^{(1)}$ are the basis of the TPCA for the 1-mode unfolded image $f_{(1)}$. After 2-mode unfoldings, $u_j^{(2)}, u_{j'}^{(2)}$ and $u_{j''}^{(2)}$ are the basis of the TPCA for the 2-mode unfolded image $f_{(2)}$. $d_{ij}, d_{i'j'}, d_{i''j''}$ are the basis of the 2DDCT. After the PCA for the vectorised image, $u_k, u_{k'}, u_{k''}$ are the basis of the PCA. After the PCA for the 1- and 2-mode unfolded image, $u_{ij}, u_{i'j'}, u_{i''j''}$ are the basis of the 2D tensor space. Here, $u_{ij} = u_i^{(1)} \otimes u_j^{(2)}$, $u_{i'j'} = u_{i'}^{(1)} \otimes u_{j'}^{(2)}$ and $u_{ij} = u_{i''}^{(1)} \otimes u_{j''}^{(2)}$. By selecting the basis, we obtain an orthogonal projection to a lower-dimensional subspace. 58

2.9 Volumetric image representation and dimension reduction. In the origin of the flow, $e_{ijk}, e_{i'j'k'}, e_{i''j''k''}$ are the basis representing each pixel of an image f. After vectorisation of the image, $e_l, e_{l'}, e_{l''}$ are the standard basis for the Euclidean space for the vectorised image $\mathrm{vec}\, f$. After 1-mode unfolding, $u_i^{(1)}, u_{i'}^{(1)}$ and $u_{i''}^{(1)}$ are the basis of the TPCA for the 1-mode unfolded image $f_{(1)}$. After 2-mode unfoldings, $u_j^{(2)}, u_{j'}^{(2)}$ and $u_{j''}^{(2)}$ are the basis of the TPCA for the 2-mode unfolded image $f_{(2)}$. After 3-mode unfoldings, $u_k^{(3)}, u_{k'}^{(3)}$ and $u_{k''}^{(3)}$ are the basis of the TPCA for the 3-mode unfolded image $f_{(3)}$. $d_{ijk}, d_{i'j'k'}, d_{i''j''k''}$ are the basis of the 3DDCT. After the PCA for the vectorised image, $u_l, u_{l'}, u_{l''}$ are the basis of the PCA. After the PCA for the 1-, 2- and 3-mode unfolded images, $u_{ijk}, u_{i'j'k'}, u_{i''j''k''}$ are the basis of the 3D tensor space. Here, $u_{ijk} = u_i^{(1)} \otimes u_j^{(2)} \otimes u_k^{(3)}$, $u_{i'j'k'} = u_{i'}^{(1)} \otimes u_{j'}^{(2)} \otimes u_{k'}^{(3)}$, $u_{i''j''k''} = u_{i''}^{(1)} \otimes u_{j''}^{(2)} \otimes u_{k''}^{(3)}$, By selecting the basis, we obtain an orthogonal projection to a lower-dimensional subspace. 63

3.1 Differences in the dimension-reduction path among downsampling, the pyramid transform, the two-dimensional discrete cosine transformation, the two-dimensional random projection, the random projection and multidimensional scaling. After the sampling of an original image, dimension-reduction methods mainly follow two paths. In the first path, after the reduction of the image, it is converted to a vector. In the second path, after vectorisation, the dimension of the feature vector is reduced. Here, $m, m', n, n', d, k \in \mathbb{Z}$ and $n' < n, m' < m, k < d$. 76

3.2 Angle between two functions and a nonexpansive map. $f, g \in H$ are functions and ϕ is a nonexpansive mapping. Here, $\angle(f, g)$ represents the angle between f and g. 83

3.3 (a) Random projection (RP). Let $\boldsymbol{x}_i \in X$ be a point and $\hat{\boldsymbol{x}}_i = \boldsymbol{R}\boldsymbol{x}_i$. The distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is preserved in the projected space $\mathbb{R}^k$. (b) Preservation of angles and volumes. Points $\boldsymbol{x}_i, \boldsymbol{x}_j$ and $\boldsymbol{x}_k$ are in the original space, and points $\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j$ and $\hat{\boldsymbol{x}}_k$ are in the projected space. The RP preserves the angle θ in the projected space $\mathbb{R}^k$ as $\hat{\theta}$. The grey regions illustrate the area of a triangle. The RP also preserves areas and volumes. (c) Preservation of manifolds. The curved plane with dashed lines illustrates a manifold in the original space $\mathbb{R}^d$. The solid lines illustrate the projected manifold in $\mathbb{R}^d$. (d) Differences in two RP paths. 86

3.4 Requirements for a mapping in image pattern recognition. (a) Order condition. For classification including categorisation, dimension reduction should preserve the order structure in the original space. However, in many cases, data are embedded in a metric space and classified with respect to a metric. For example, in visual categorisation, histograms of images are embedded in a metric space. (b) Weak condition. The dimension reduction operation Φ approximately preserves distances and angles among data. (c) Strong condition. Φ preserves distances and angles among data. Only the rotation transform satisfies this condition. (d) Discriminative condition. Φ locally shrinks neighbourhoods and globally expands distances and angles among elements in the data space. This mapping increases the separation ratio between distributions of categories, although it is not a dimension reduction mapping. . . 90

3.5 (a) Geometric properties of the subspace method (SM). Let $\boldsymbol{\varphi}_1$ and $\boldsymbol{\varphi}_2$ be the bases of a class pattern. For an input $\boldsymbol{g}$, similarity is defined as the orthogonal projection to the pattern space. (b) Multiclass recognition using the SM. Let $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ be operators for subspaces $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. The input $\boldsymbol{g}$ is labelled as being in the 1st class since subspace $\mathcal{C}_1$ has the longer projection length of $\boldsymbol{g}$. (c) Angle between two linear subspaces $\mathcal{C}_1$ and $\mathcal{C}_2$. The minimal angle between the two subspaces is 0. However, in the mutual subspace method (MSM), we adopt the angle θ to indicate the similarity between two subspaces. (d) Multiclass recognition using the MSM. For an input subspace $\mathcal{C}_{\boldsymbol{g}}$, let θ_1 and θ_2 be its angles relative to $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. The input subspace $\mathcal{C}_g$ is labelled as being in the 1st class since $\theta_1 < \theta_2$. (e) Projection onto constraint subspace. The triangles represent the subspace of categories $\mathcal{C}_1$ and $\mathcal{C}_2$ and the subspace of queries $\mathcal{C}_{\boldsymbol{g}}$. The left figure shows three subspaces in the pattern space $\mathbb{R}^d$. The right figure shows the subspaces in the constraint subspace $\mathcal{D}_k$. In the constraint subspace, the relation between $\mathcal{C}_1$ and $\mathcal{C}_2$ ideally becomes orthogonal since we omit the common subspace between subspaces $\mathcal{C}_1$ and $\mathcal{C}_2$. 95

3.6 Examples of three categories in each dataset. (a) YaleB. (b) ORL. (c) ETH80. (d) NEC. (e) MNIST. (f) ETL9G. (g) CALTECH101. (h) VOC2012. 109

3.7 Energy loss of dimension-reduction methods. (a)-(f) show the energy loss for the pyramid transform, downsampling, random projection, two-dimensional random projection, two-dimensional discrete cosine transform and metric multidimensional scaling, respectively. In these figures, circles, upward triangles, downward triangles, squares, diamonds, asterisks, five-pointed stars and six-pointed stars represent the YaleB, ORL, ETH80, NEC, MNIST, ETL9G, CALTECH101 and VOC2012 datasets, respectively. The horizontal and vertical axes represent the compression ratio and energy loss, respectively. . . . 110

3.8 Mean relative error between distances in original space and dimension-reduced space. In the computation, we randomly select 1000 pairs from each dataset. In (a)-(h), circles, upward triangles, downward triangles, squares, diamonds and asterisks represent the pyramid transform, downsampling, random projection, two-dimensional random projection, two-dimensional discrete cosine transform and metric multidimensional scaling, respectively. The horizontal and vertical axes represent the compression ratio and relative error, respectively. 111
3.9 Cumulative contribution ratios. In (a)-(h), stars, circles, upward triangles, downward triangles, squares, diamonds and asterisks represent the cumulative contribution ratios of the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection, two-dimensional discrete cosine transform and metric multidimensional scaling, respectively. The horizontal and vertical axes represent the compression ratio and cumulative contribution ratio, respectively. The cumulative contribution ratio is displayed as a logarithm to base 10. 112
3.10 Convergences of full projection method and full projection truncation method. In (a)-(h), circles and squares represent the sum of energies of all projected images in each step of the iteration for the full projection (FP) method and full projection truncation (FPT) method, respectively. Horizontal and vertical axes present the number of iterations and the sum of energies of all projected images, respectively. 113
3.11 Cumulative contribution ratios. In (a)-(b), circles, downward triangles and squares represent the cumulative contribution ratio of the eigevalues of mode 1 for the full projection (FP) method, full projection truncation (FPT) method and marginal eigenvector (MEV) method, respectively. In (a)-(b), upward triangles, leftward triangles and diamonds represent the cumulative contribution ratio of the eigenvalues of mode 2 for the FP method, FPT method and MEV method, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and cumulative contribution ratio, respectively. The cumulative contribution ratio is displayed as a logarithm to base 10. The compression ratio is the ratio to the original size of the images in Table 3.2. 114

3.12 Recognition rates for each pair consisting of dimension-reduction method and classification method in the YaleB dataset. Stars, circles, upward triangles, downward triangles, squares, diamonds and asterisks represent the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection, two-dimensional discrete cosine transform and metric multidimensional scaling, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively. 115
3.13 Recognition rates for each pair consisting of dimension-reduction method and classification method in the ORL dataset. Stars, circles, upward triangles, downward triangles, squares, diamonds and asterisks represent the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection, two-dimensional discrete cosine transform and metric multidimensional scalng, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively. 116
3.14 Recognition rates for each pair consisting of dimension-reduction method and classification method in the ETH80 dataset. Stars, circles, upward triangles, downward triangles, squares, diamonds and asterisks represent the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection, two-dimensional discrete cosine transform and metric multidimensional scaling, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively. 117
3.15 Recognition rates for each pair consisting of dimension-reduction method and classification method in the NEC animal dataset. Stars, circles, upward triangles, downward triangles, squares, diamonds and asterisks represent the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection, two-dimensional discrete cosine transform and metric multidimensional scaling, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively. 118

3.16 Recognition rates for each pair consisting of dimension-reduction method and classification method in the MNIST dataset. Stars, circles, upward triangles, downward triangles, squares and diamonds represent the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection and two-dimensional discrete cosine transform, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively. 119
3.17 Recognition rates for each pair consisting of dimension-reduction method and classification method in the ETL9G dataset. Stars, circles, upward triangles, downward triangles, squares and diamonds represent the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection and two-dimensional discrete cosine transform, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively. 120
3.18 Recognition rates for each pair consisting of dimension-reduction method and classification method in the CALTECH101 dataset. Stars, circles, upward triangles, downward triangles, squares and diamonds represent the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection and two-dimensional discrete cosine transform, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively. 121
3.19 Recognition rates for each pair consisting of dimension-reduction method and classification method in the VOC2012 dataset. Stars, circles, upward triangles, downward triangles, squares and diamonds represent the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection and two-dimensional discrete cosine transform, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively. 122

3.20 Recognition rates for the vector-representation-based subspace method. In (a)-(h), stars, circles, upward triangles, downward triangles, squares and diamonds represent the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection and two-dimensional discrete transform, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively. 123

3.21 Recognition rates of the matrix-representation-based tensor subspace method. In (a)-(h), stars, circles, upward triangles, downward triangles, rightward triangles, leftward triangles, squares and diamonds represent the original dimension, pyramid transform, downsampling, marginal eigenvector method, full projection method, full projection truncation method, two-dimensional random projection and two-dimensional discrete transform, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively. 124

4.1 Mathematical properties of expression of images for computation of the distances between images. (a) Gray-scale images. We generally use L_2-norm for them as a distance. representation. (b) Probabilistic distribution of gray values in images. Images are represented by probabilistic distributions by normalised as there L_2-norms to be 1. Wasserstein distance is defined by the sum of transportation costs between two probabilistic distributions. (c) Decomposition of images by tensor principal component analysis. Images are decomposed to eigenvalues and eigenvectors. Wasserstein distance is defined by the sum of transportation costs for contribution ratios of eigenvalues. In the transportation, an angle between bases is adopted as a cost for transportation. 132

4.2 The Wasserstein distance between tensor subspaces for second-order tensors. Tensor subspaces are obtained by tensor principal component analysis for each given image. Note that a tensor subspace is obtained from an image. For these subspaces, transportation between contribution ratios of eigenvalues is considered. (a) and (b) show contribution ratios of eigenvalues obtained by singular value decomposition for different two images. An angle between bases are computed as a transportation cost between eigenvalues. Wasserstein distance is the result of the minimisation of total transportation cost among eigenvalues for two tensor subspaces. 134

4.3 Examples of sequences of silhouette images. The figures are gait images whose pixel values are 0 or 255. The figure illustrate the 1st, 21st, 41st, 61st, 81st silhouette images of sequences from a person walking at different speeds. Each sequence consists of 90 silhouette images of four steps. For each sequence, we manually selected the start and finish of the sequence from the original OU-ISIR treadmill dataset. Each sequence is obtained by resampling of the selected sequence with linear interpolation, where the linear interpolation is only used for mode 3. 136

4.4 Convergence of iteration described in Algorithm 1. (a)-(c) show the sum of energies Ψ_k in each iteration for the given numbers of bases of $128 \times 88 \times 90$, $64 \times 64 \times 64$ and $32 \times 32 \times 32$, respectively. In the (a)-(c), horizontal and vertical axes represent the number of iterations and Ψ_k, respectively. For the computation of the tensor projections using Algorithm 1, we adopt the six orders of selection of the modes, where the legends in the figures summarises the six orders. 138

4.5 Cumulative contribution ratio of eigenvalues obtained by 10 iterations using Algorithm 1. (a)-(c) show the cumulative contribution ratios for the 1-, 2- and 3-modes, respectively. Here, the given number of bases is $128 \times 88 \times 90$ for Algorithm 1. For the computation of the tensor projections using Algorithm 1, we adopt six orders of the selection of modes, where the legends in the figures summarises the six orders. The horizontal and vertical axes represent the compression ratio and cumulative contribution ratio, respectively. For the original size $D = 128 \times 88 \times 90$ and the reduced size $K = k \times k' \times k''$, the compression ratio is given as D/K. 138

4.6 Cumulative contribution ratio of three modes. (a)-(c) show the cumulative contribution ratios of the three modes for the input sizes $128 \times 88 \times 90$, $64 \times 64 \times 64$ and $34 \times 34 \times 34$ in Algorithm 1, respectively. The horizontal and vertical axes represent the compression ratio and cumulative contribution ratio, respectively. For the original size $D = 128 \times 88 \times 90$ and the reduced size $K = k \times k' \times k''$, the compression ratio is given as D/K. 140

4.7 Comparison of cumulative contribution ratio between full projection and full projection truncation. (a)-(c) show a comparison of the cumulative contribution ratio in the three modes. The horizontal and vertical axes represent the compression ratio and cumulative contribution ratio, respectively. For the original size $D = 128 \times 88 \times 90$ and the reduced size $K = k \times k' \times k''$, the compression ratio is given as D/K. 140

4.8 Comparison of cumulative contribution ratio for three types of compressed tensor. For the compression of tensors, we use Algorithm 1 and the 3DDCT. In Algorithm 1, we respectively adopt sizes of $128 \times 88 \times 90$ and $32 \times 32 \times 32$ for the computation by FP and FPT. For the three types of compressed tensor of $32 \times 32 \times 32$, we apply 10 iterations of Algorithm 1. In (a)-(c), the horizontal and vertical axes represent the compression ratio and cumulative contribution ratio, respectively. For the first reduced size $K_1 = 32 \times 32 \times 32$ and the second reduced size $K_2 = k \times k' \times k''$, the compression ratio is given as K_1/K_2. 141

4.9 Recognition rates of gait patterns and liver data for original and compressed tensors. We adopt the reduces sizes of $32 \times 32 \times 32$, $16 \times 16 \times 16$ and $8 \times 8 \times 8$. (a)-(c) show the recognition rates for three reduced sizes of OU-ISIR. For compression, we use the HOSVD, FP, FPT and 3DDCT. In (a)-(c), the horizontal and vertical axes represent the compression ratio and recognition ratio [%], respectively. In (a)-(c) for the original reduced sizes $D = 128 \times 88 \times 90$, the compression ratio is given as D/K for reduced size k. 142

4.10 Computational time of dimension reduction for tensors of the order three. (a) and (b) show the computational time of construction of projection matrices for 306 sequences of silhouette images and 35 voxel images of livers, respectively . (c) shows the mean computational time of projecting images to low-dimensional tensor space for OU-ISIR and CA datasets. In (a) and (b), we compare the HOSVD, FP, FPT and 3DDCT. In (a)-(c), the vertical and horizontal axes represent the computational time and compression ratio, respectively. 142
4.11 Original and reconstructed volumetric data of liver data. (a) shows the rendering of original data. (b)-(d) show the rendering of reconstructed data after the FP, FPT and 3DDCT, respectively. (e)-(f) illustrate axial slice images of these volumetric data in (a)-(d), respectively. The sizes of reduced tensors are shown in Table. 1. 144
4.12 Cumulative contribution ratios for three compressed tensors. For compression, we adopt FP, FPT and 3D-DCT. For the computation of the cumulative contribution ratio of eigenvalues obtained by the FP, we used all eigenvalues of modes 1, 2 and 3 after sorting them into descending order. 145
4.13 Reconstruction by using only major principal components of the decomposition by the FP. Top and bottom rows illustrate volume rendering and axial slice of reconstructed data, respectively. For reconstruction, we use the 20 major principal components. Left, middle and right columns illustrate the results for the tensors projected by the FP, FPT and 3D-DCT. . . . 146
4.14 (a) and (b) illustrate the examples of livers of male and female, respectively. 146
4.15 Recognition rates of liver data for original and compressed tensors. For compression, we use the HOSVD, FP, FPT and 3D-DCT. The horizontal and vertical axes represent the compression ratio and recognition ratio [%], respectively. For the original size $D = 89 \times 97 \times 76$ and the reduced size $K = k \times k' \times k''$, the compression ratio is given as D/K for reduced size k. . . 147

4.16 Illustration of extracted cardiac MRI dataset. These sequences of volumetric data are extracted from cardiac MRI dataset with landmarks of endocardium of left ventricles [10]. As shown in Table 1, we have 17 sequences of volumetric data of left ventricle for 17 patients. Each sequence of volumetric data represents one cardiac beat by 20 frames. Every sequence starts with maximally expanded state. Red and white parts of volume rendering of the data represent muscle and inner space of left ventricles. We set the center of the first sagittal slice of each volume data to the center of the slice. 149
4.17 Shape and inner texture of reconstructed volume data of left ventricle from compressed data. Upper and lower rows show volume rendering and sagittal slice of the volumetric data, respectively. In (a)-(d), red and white parts depict the muscle of heart and inner of heart, respectively, for original and approximation by the FP, the FPT and the 3D-DCT. In these approximation, the data are reduced to the size $16 \times 16 \times 16$. 150
4.18 Extracted principal components of dimension-reduced volume data. For the data dimension reduced by the FP, FPT and 3D-DCT, we apply the FP. Using the extracted principal component, we reconstruct volumetric data. For the extraction, we select the 20 principal eigenvectors of ones of three modes. 151
4.19 Recognition rates of the left ventricles for original and compressed tensors. We use tensor subspace method as classifier. The data are reduced to $32 \times 32 \times 32$, $16 \times 16 \times 16$ and $8 \times 8 \times 8$. The HOSVD, FP, FPT and 3D-DCT are used for the reduction. Vertical and horizontal axes represent recognition rate and compression ratio, respectively. For the original size $D = 81 \times 81 \times 63$ and reduced size $K = k \times k' \times k'$, the compression ratio is given by D/K. 152

4.20 Recognition rates of left ventricles for compressed tensors. We adopt the reduces sizes of $32 \times 32 \times 32$, $16 \times 16 \times 16$ and $8 \times 8 \times 8$. For compression, we use the HOSVD, FP, FPT and 3D-DCT. In the mutual tensor subspace method, input is a query subspace. The query subspace is spanned by a few queries. To construct a query subspace, we use one, two and three queries. Top, middle and bottom row show recognition rates for the case of one, two and three queries, respectively. Vertical and horizontal axes represent recognition rate and compression ratio, respectively. For the original size $D = 81 \times 81 \times 63$ and reduces size $K = k \times k' \times k'$, the compression ratio is given by D/K. 153
4.21 Silhouette images of a walking person in OU-ISIR dataset. The top, middle and bottom row represent sequence of four steps in different speeds for a man. 154
4.22 Wasserstein distances and Euclidean distance between first and ith frames for $i = 1, 2, \ldots, 90$. Plotted Euclidean distance is the relative distance for the L_2-norm of the first frame. This relative distance is defined by $\|\boldsymbol{X}_1 - \boldsymbol{X}_i\|_{\mathrm{F}}/\|\boldsymbol{X}_1\|_{\mathrm{F}}$, where $\boldsymbol{X}_1$ and $\boldsymbol{X}_i$ are the first and ith frames, and $\|\cdot\|_{\mathrm{F}}$ is Frobenius norm. 155
4.23 Example of decomposition of an image. (a) The first frame of walking person in 2km/h of OU-ISIR treadmill dataset. (b) contribution ratio of eigenvalues obtained by singular value decomposition. (c) cumulative contribution ratio of eigenvalues obtained by singular value decomposition 156
4.24 Reconstruction of the first frame. In (a), (b) and (c), the first frame reconstructed by using one, three, six major eigenvectors for mode-1 and -2. In (d), (e) and (f),the first frame reconstructed by using 30, 28 and 25 minor eigenvectors for mode-1 and -2. 157
4.25 Successive two frames in a sequence. (a) Pre frame. (b) Post frame. (c) The difference of two frames. For visualisation, each pixel of the difference is displayed in its absolute value. . 158
4.26 Absolute value of inner products for eigenvectors between 1st and 2nd frames. (a) and (b) show the inner products for eigenvectors of mode 1 and 2, respectively. In (a) and (b), from top to bottom, rows represent the eigenvectors of the second frame in descending order of eigenvalues. In (a) and (b), from left to right, columns represent the eigenvectors of the first frame in descending order of eigenvalues. 158

4.27 Difference between reconstructed images. 159
4.28 Wasserstein distances between first and ith frames for major and minor principal components. 160
4.29 Wasserstein distances for mode-1 and -2. Top and bottom rows summarise the Wasserstein distances by using major and minor principal components, respectively. 162
4.30 Wasserstein distances between substances of the first and kth categories. For the computation of the distances, we use 64, 57 and 20 eigenvectors of a category's tensor subspace for 1, 2 and 3 modes, respectively. 163
4.31 A first frame from the sequence reconstructed from principal components obtained by third-order principal component analysis. (a) reconstructed frame from one major principal component of each mode. (b) reconstructed frame from four major principal components of each mode. (c) reconstructed frame from ten major principal components of each mode. . . 163
4.32 Wasserstein distances for sets of major and minor eigenvectors. (a), (c) and (e) show the Wasserstein distances between sets of major eigenvectors. (b), (d) and (f) show the Wasserstein distances between sets of minor eigenvectors. 164
4.33 Sequence of beating heart. (a)-(t) show a sequence of a beating heart. 165
4.34 Wasserstein distances and Euclidean distance between first and ith frames for $i = 1, 2, \ldots, 20$. Plotted Euclidean distance is the relative distance for the L_2-norm of the first frame. T This relative distance is defined by $\|\boldsymbol{X}_1 - \boldsymbol{X}_i\|_{\mathrm{F}} / \|\boldsymbol{X}_1\|_{\mathrm{F}}$, where $\boldsymbol{X}_1$ and $\boldsymbol{X}_i$ are the first and ith frames, and $\|\cdot\|_{\mathrm{F}}$ is Frobenius norm. 166
4.35 Example of an decomposition of an image. (a) first frame. (b) contribution ratio of eigenvalues. (c) cumulative contribution ratio of eigenvalues. 167
4.36 Reconstruction of the first frame. In (a), (b) and (c), the first frame is reconstructed by using one, three, six major eigenvectors for mode-1 and -2. In (d), (e) and (f), the first frame is reconstructed by using 52, 50 and 47 minor eigenvectors for mode-1 and -2. 169
4.37 Successive two frames in a sequence. (a) Pre frame. (b) Post frame. (c) The difference of two frames. For visualisation, each pixel of the difference is displayed in its absolute value. . 170

4.38 Absolute value of inner products for eigenvectors between 1st and 2nd frames. (a) and (b) show the inner products for eigenvectors of mode 1 and 2, respectively. In (a) and (b), from top to bottom, rows represent the eigenvectors of the second frame in descending order of eigenvalues. In (a) and (b), from left to right, columns represent the eigenvectors of the first frame in descending order of 53 eigenvalues. 171
4.39 Difference between reconstructed images. 172
4.40 Wasserstein distances between first and ith frames for major and minor principal components. 172
4.41 Wasserstein distances for mode-1 and -2. Top and bottom rows summarise the Wasserstein distances computed by using major and minor principal components, respectively. 173
4.42 Wasserstein distances between first and ith frames for major and minor principal components. 174
4.43 Wasserstein distances for major and minor eigenvectors. (a), (c) and (e) show the Wasserstein distances between sets of major eigenvectors. (b), (d) and (f) show the the Wasserstein distances between sets of minor eigenvectors. 175
4.44 Wasserstein distances between first and ith frames computed by using 3rd-order-tensor representation. 176
4.45 The first frames from the sequences reconstructed from principal component obtained by third-order principal component analysis. (a), (b) and (c) are the first frames of the sequence reconstructed by one, four and ten major principal components, respectively. 176
4.46 Volume rendering of reconstructed and original sequences. In (a)-(d), faces at left down part of images, on which white part exist, are the first frame of sequences. That is, these part in (a), (b), (c), and (d) coincident to the images shown in Figs. 4.36(a), (b) and (c), and Fig. 4.35(a), respectively. From the left down part to right up part, slices of a sequence are placed in times series. For these display, the voxel size for the direction of times series are multiplied by four since the data array is $81 \times 81 \times 20$ voxels. For these rendering, voxel size for time axis are multiplied by 4. 177
4.47 Wasserstein distances for mode-1 and -2. Top and bottom rows summarise the Wasserstein distances by using major and minor principal components, respectively. 178

5.1 Pipeline of the HoG method. The pipeline consists of three steps: feature extraction, selection of metric and classifier. At the first step of feature extraction, the HoG method generate signature from distribution of gradients in local regions. At the second step of selection of metric, the HoG method adopts L_2-norm. At the third step of selection of classifier, the HoG method adopts kernel support vector machine (SVM). In this step, the HOG method needs appropriate kernel design for kernel SVM for a problem. 183
5.2 Example of directional statistics. (a) Grayscale image. (b) Magnitudes of gradients in a local region of the image. (c) Directions of gradients in a local region of the image. (d) Distribution of structure tensors in a local region of the image. (e) Circular histogram constructed with the gradient field. . . 189
5.3 Rotation invariance in shape of circular histogram. (a) Original image $f(\boldsymbol{x})$ and rotated image $f(\boldsymbol{y})$. (b) Histogram $h(\theta)$ obtained from the original image. (c) Histogram $h'(\theta)$ obtained from the rotated image. The histogram in (c) is the histogram in (b) after rotation. 191
5.4 Methods of aggregating local regions and of measuring difference between aggregated local regions. (a) Whole region of an image with no division. (b) Cells dividing an image. (c) Cells aggregated into blocks. As shown in (d)-(f), we have three methods to discriminate the difference between two images. (d) Difference between images. (e) Sum of differences between cells. (f) Sum of differences between blocks. 192
5.5 Flow of feature extraction methods. There are three flows. The top row shows feature extraction from the entire region of an image. The middle and bottom rows show feature extraction from cells and blocks, respectively. In the middle and bottom rows, a set of histograms is extracted. For these three extraction methods, we can adopt three different types of histograms. In other words, we can use the simple directional distribution, the directional distribution and the dominant directional distribution. The small box on the left summarises the extraction of the dominant directional distribution in each cell. 196

5.6 Feature extraction of the HoG method. As shown in (a), an image is divided into cells C_{ij} and the DDs of each local region $B_k, k \in \{1, 2, 3, 4\}$ are obtained by a moving window for $i, j \in \{1, 2, 3\}$. (b) shows how to construct a feature vector in the HOG method. Extracted histograms in each cell C_{ij} are represented as column vectors $\boldsymbol{h}_{i'j'}$. The vectorised histograms in each block are connected and normalised by the ℓ_2-norm. By connecting these ℓ_2-normalised vectors, we obtain the feature vector. 199

5.7 (a) Set of 38 positive images for deciding a median. (b) Examples of positive queries of pedestrians. The set of positive queries does not include the set of 38 images in (a). The total number of positive queries is 115. (c) Examples of negative queries of pedestrians. The total number of negative queries is 115. All images have a resolution of 130×70 pixels. For feature extraction, we use only the centre region of 124×64 pixels of these images. 204

5.8 Recognition rates and ROC curves for the simple directional distribution (SDD) feature, directional distribution (DD) feature and histogram of oriented gradients (HoG) feature. Left and right columns show the results for the recognition rate and ROC curve, respectively. Top, middle and bottom rows show results for the SDD, DD and HoG features, respectively. In (a), (c) and (e), the vertical and horizontal axes represent the recognition rate and criterion, respectively. In (b) (d) and (f), the vertical and horizontal axes represent the true positive rate and false positive rate for each given criterion, respectively. The discrimination method using the L_1-norm gives the highest recognition for the SSD, DD and HoG features. . 207

5.9 Recognition rates and ROC curves for the global DD features. Upper and lower rows respectively show the recognition rates and ROC curves. The first, second, third and fourth columns show the results for discrimination using the L_1-norm, L_2-norm, 1-Wasserstein distance (1WD) and binomial-distribution-based 1-Wasserstein distance (B1WD), respectively. In (a)-(d), the vertical and horizontal axes represent the recognition rate and criterion, respectively. In (e)-(h), the vertical and horizontal axes represent the true positive rate and false positive rate, respectively. In (a)-(d), circles, squares, six-rayed stars, and upward and downward triangles represent results for blurred images with Gaussian filtering with standard deviations of $0, 2, 4, 8$ and 16, respectively. 208

5.10 Recognition rates and ROC curves for the global DDD features. Upper and lower rows respectively show the recognition rates and ROC curves. The first, second, third and fourth columns show the results for the discrimination using the L_1-norm, L_2-norm, 1-Wasserstein distance (1WD) and binomial-distribution-based 1-Wasserstein distance (B1WD), respectively. In (a)-(d), the vertical and horizontal axes represent the recognition rate and criterion, respectively. In (e)-(h), the vertical and horizontal axes represent the true positive rate and false positive rate, respectively. In (a)-(d), circles, squares, six-rayed stars, and upward and downward triangles represent the results for blurred images with Gaussian filtering with standard deviations of $0, 2, 4, 8$ and 16, respectively. . . . 209

5.11 Recognition rates and ROC curves for the DDD features. In (a), the vertical and horizontal axes represent the recognition rates and criteria, respectively. In (b), the vertical and horizontal axes represent the true positive rate and false positive rate for each given criterion, respectively. 210

5.12 Distribution of L_1-norms among the median and queries. (a) Relation between L_1-norms for L_1- and L_2-normalised HoG features. The horizontal and vertical axes represent L_1-norms for L_1- and L_2-normalised HoG features, respectively. This relation shows that the mapping ϕ is nonlinear. In (b) and (c), the horizontal and vertical axes represent the distances between the median and the queries and their probability of occurrence, respectively. L_2-normalisation gives more discriminative distributions for positive and negative queries than L_1-normalisation. 211

6.1 (a) Nearest neighbours of g searched for by the k-nearest-neighbour search on a manifold. Our method of projecting the manifold to a low-dimensional subspace. (b) Generation of a new entry in a dictionary. The input image g is projected onto the subspace spanned by three nearest neighbours. (c) Interpolation of parameter. For the new entry g^*, we interpolate the parameter θ^* of the image g^*. Here, Π represents the parameter space of the transform. 217
6.2 Registration problems in the computer vision. (a) Registration to one of the stored images. (b) Registration to the local region of an image. (c) Registration based on the camera model. . . 220
6.3 Example of two paths for line integral. (a) and (b) show circular paths and line paths, respectively. In (a), the solid line and dashed lines represent circles $\mathcal{C}_i$ and $\mathcal{C}_j$, respectively. Here, we set $r_i < r_j$. In (b), the solid and dashed lines represent $\mathcal{R}_0$ and $\mathcal{R}_5$, respectively. 228
6.4 Surfaces used for surface integration to obtain independent equations. (a) and (b) show surface $\mathcal{S}_3(r)$ of the sphere and surface $\mathcal{P}_3(r, 0)$ comprising three planes. Integration of the volume gives a equation for an image. The integration of different surfaces, such as a different spheres and orthogonal square planes, gives several independent equations for an image. . . 230
6.5 Slice images and character images. (a)-(c) are slice images extracted from volume data obtained by MRI simulation of a human brain [37]. The size of the volume data is 181×217×181 pixels. The slice images (a), (b) and (c) are extracted from the $z = 50$, $z = 48$ and $z = 52$ planes, respectively. (d)-(f) are character images in a handwriting dataset [142]. The size of the character images is 127×128. In experiments, we embed (a)-(c) and (d)-(f) in 543×543 pixel and 272×272 pixel background images, respectively. The intensities of background images are 0. 231
6.6 Accuracy of estimation of single transform for transformed slice images of human brain. Thee upper and lower rows represent the accuracy for rotation and scaling, respectively. The first column shows the accuracy for the transformed Fig. 6.5(a) with no filtering. The second, third and fourth columns show accuracy for the transformed Figs. 6.5(a), (b) and (c) with Gaussian filtering of the standard deviation τ. 232

6.7 Accuracy of estimation of single transform for transformed character images. The upper and lower rows represent the accuracy for rotation and scaling, respectively. The first column shows the accuracy for the transformed Fig. 6.5(d) with no filtering. The second, third and fourth columns show the accuracy for the transformed Figs. 6.5(d), (e) and (f) with Gaussian filtering of standard deviation τ. 233

6.8 Accuracy of estimation of multi transform for slice images and character images. The upper and lower rows represent the accuracy for rotation and scaling, respectively. From left to right, each column represents the accuracy of parameter for the transformed Figs. 6.5(a), (b), (c), (d), (e) and (f). For the radius r of the integration path, rotation angle θ and scaling factor λ, we define the displacement as $\sqrt{(1+\lambda)^2 r^2 + \lambda^2 r^2}$. . 235

6.9 Slice images extracted from volumetric data. (a)-(c) Slice images extracted from a voxel image obtained by MRI simulation of a human brain [37]. The size of the voxel image is 181×217×181 voxels. The slice images (a), (b) and (c) are extracted from the $z = 45$, $x = 90$ and $y = 100$ planes, respectively. In experiments, we embed the voxel image in a background image of 308×308×308 voxels. The intensities of the background images are 0. 236

6.10 Volumetric spatiotemporal MRI lung data [25]. (a) Voxel image of a frame of a sequence. (b)-(d) Sagittal slices of the frame. The spatial and time resolutions of the data are 50×224×224 and 200, respectively. The time between frames is 331 ms. In the experiments, we embed a volumetric image of a frame on a background image of 316×316×316 voxels. Each voxel value in the background image is 0. 237

6.11 Accuracy of estimation for a spatial rotation. We estimate the rotation angles ϕ_1, ϕ_2 and ϕ_3 independently. The first, second and third columns represent the accuracy of estimation for rotation around the x, y and z axes, respectively. (a) and (d), (b) and (e), and (c) and (f) show the accuracy of estimation without Gaussian filtering. (g) and (j), (h) and (k), and (i) and (l) show the accuracy of estimation for smooth images, for the rotation around x, y and z axes, respectively. In the first and third rows and the second and fourth rows, we adopt $\mathcal{S}_3(r)$ and $\mathcal{P}_3(r, \boldsymbol{\phi})$ as the surfaces for the surface integration, respectively. Displacements are given by $r\phi_1, r\phi_2$ and $r\phi_3$. . . 238

6.12 Accuracy of estimation for multiple transforms. For the estimation, we adopt combinations of three rotations around the x, y and z axes. The left, middle and right graphs show the results of estimation for rotation around the x, y and z axes with Gaussian filtering with standard deviation τ, respectively. For the surface integration, we adopt surfaces $\{\mathcal{P}_i^1\}_{i=1}^n$. For the rotations around the x, y and z axes, the displacements are given by $r\sqrt{\phi_3^2+\phi_2^2}$, $r\sqrt{\phi_3^2+\phi_1^2}$ and $r\sqrt{\phi_1^2+\phi_2^2}$ with radius r in the surface integral, respectively. 239

6.13 Estimation for a template with small pattern perturbation. (a) Difference between 22nd frame and 23rd-200th frames of four-dimensional MRI lung data. (b) Scaled-up graph of (a) showing difference between 22nd frame and 23rd-38th frames. (c) Accuracy of estimation for rotation angle ϕ_3 around the z axis. The differences between the 22nd frame and the 22nd, 23rd, 24th, 25th and 34th frames are $-\infty$, -10.11, -9.19, -6.94 and -5.65 [dB], respectively. For surface integration, we adopt the surface $\mathcal{S}_3(r)$. The displacement is given by $r\phi_3$. 241

List of Tables

3.1 Glossary of abbreviations. 77
3.2 Details of each database. ♯class and ♯data/class represent the number of classes and the number of data in each class, respectively. The image size is the original size of the images in each dataset. The vectorised size is the size of the vectorised images. The reduced dimension is the dimension of the images after vector-representation-based dimension reduction. The reduced image size is the size of the images after image-representation-based dimension reduction. In the CALTECH101 and VOC2012, each image has a different resolution and aspect ratio. For evaluation, we downsampled images in the CALTECH101 and VOC2012 to 92×80 pixels and 111×142 pixels, respectively. 99
3.3 Conditions of images in each dataset. This table summarises whether an image in each dataset includes cropping of the region of interest, centring of the target in the image, changes in illumination, changes in camera position and the same background. Furthermore, the last term "same object" shows whether images are taken of the same object. ◯ indicates the satisfaction of a condition. △ indicates that a condition is partly satisfied in a dataset. × indicates the nonsatisfaction of a condition. 100

3.4 Summary of evaluation of dimension-reduction methods. For the evaluation of the energy loss and relative error, ◯, △ and × represent preservation with a small compression ratio, preservation and no preservation, respectively. For the evaluation of the cumulative contribution ratio, we give some remarks. In this table, DS, PT, RP, 2DRP, 2DDCT and MDS are abbreviations for the downsampling, pyramid transform, random projection, two-dimensional random projection, two-dimensional discrete cosine transform and multidimensional scaling, respectively. 101
3.5 Dimensions of the class subspaces used in the classification. ♯query represents the number of queries for each category. ♯basis represents the number of bases used for each category in recognition. The dimension of the class subspace represents the dimension of the constraint subspace. 102
3.6 Summary of results of recognition rates. In this table, SM, MSM, CMSM and 2DTSM are abbreviations for the subspace method, mutual subspace method, constraint mutual subspace method and two-dimensional tensor subspace method, respectively. 104

4.1 Glossary of abbreviations. 126
4.2 Sizes and number of tensors of the resampled OU-ISIR.♯class and ♯data/class represent the number of classes and the number of data in each class, respectively. The tensor size is the size of the dataset before dimension reduction. The reduced tensor size is the size of the tensor after dimension reduction. We set $d \in \{32, 16, 8\}$ for the size in the dimension reduction. 137
4.3 Sizes and number of volumetric data of livers. ♯data represents the number of livers obtained from different patients. The data size is the original size of the volumetric data. The reduced data size is the size of the volume data after tensor-based reduction. 143
4.4 Sizes and number of volumetric data of left ventricles. ♯category represents the number of individuals. ♯data/category represents the number of frames in one sequence of left ventricles. The data size is the original size of the volumetric data. The reduced data size is the size of the volumetric data after reduction. We set $d \in \{8, 16, 32\}$. 148

4.5 Reconstruction error of volumetric data. The reconstruction error is given by distance between tensors of the original and reconstructed volumetric data. 150

5.1 Glossary of abbreviations. 184
5.2 Summary of histograms defined in section 3 and 4.3. 195
5.3 Summary of pairs of features and discrimination criteria. For the simple directional distribution (SDD), directional distribution (DD), dominant directional distribution (DDD) and histogram of oriented gradients (HoG), this table shows whether each discrimination method is available or not by $\bigcirc$ and $\times$, respectively. Features are divided with respect to whether they are based on a local histogram or a global histogram. 196
5.4 Details of computation of discrimination method for aggregated DD. '♯Histogram' represents the number of histograms that are used in the discrimination. '♯Direction' represents the number of quantised directions. 'Computational time' shows the average computational time of each discrimination method given by the Wasserstein distance (WD) and L_p-norm (L_p). For practical computation, we use a Xeon X5570 2.93GHz (quad core) processor. The HoG method is included in the types of blocks. 'No division' represents the global method. . 203
5.5 Summary of discriminative combinations of features and metrics. 206

6.1 Parameters for the first and second experiments using voxel image of human brain data. 236
6.2 Data for the the third experiment using the volumetric spatiotemporal data. 236
6.3 Evaluation of approximation for generated new entries. We generate new entries for rotated images with small pattern perturbation. For a generation of a new entry, we use 4-neighbours of a template. As templates, we use rotated images of the 22nd, 23rd, 24th and 25th frame of data with angle ϕ_3. For a template g, we first compute the difference between g and its nearest neighbour in pregenerated images as $10\log_{10}(\|f^1 - g\|_2/\|g\|_2)$. Second, we compute the difference between g and a generated new entry g^* as $10\log_{10}(\|g^* - g\|_2/\|g\|_2)$. In this Table, the columns for the nearest neighbour (NN) and the local linear method (LLM) show the difference between f^1 and g and between g^* and g, respectively. 240

6.4 Accuracy and compression ratio for volumetric data obtain by MRI simulation of human brain. First column shows given accuracy in the estimation. Second column shows necessary step sizes in pregeneration, which give the accuracy in first column, for the nearest neighbour search (NNS) and the local linear method. Third column shows dimensions of search space for NNS and LLM. Fourth column illustrates compression ration of the LLM compared with the NNS. 240

Chapter 1

Introduction

1.1 Background and Purpose

This dissertation aims to construct the computational methods for multiway data in pattern recognition. We extend traditional representation of patterns, which subspace method deal with in pattern recognition, to multilinear forms by adopting a tensor. Tensor forms offer multiway operation for multidimensional array. Therefore, we can use multidimensional array as multiway data in a tensor space. This extension allows us to deal with multidimensional array without vectorising. Then, we extend traditional subspace method for multiway data. Furthermore, by adopting methods for multidimensional signal processing, we construct fast and robust computational method for the extended subspace method. Finally, we apply the constructed methods to actual multiway data of problems in pattern recognition.

The recognition, detection, visual categorisation and image retrieval problems in computer vision are defined in ref. [40] by Csurka *et al.*.

- *Recognition*: The identification of instances of particular objects. For instance, recognition distinguishes between images of two structurally distinct cups, while visual categorisation places them in the same class.

- *Detection*: The process of deciding whether or not a member of a visual category is present in a given image. Most previous work on detection has centred on machine learning approaches to detecting faces, cars or pedestrians.

- *Visual Categorisation*: A process that is sufficiently generic to cope with many object types simultaneously and can be readily extended to new object types. At the same time, the process should handle variations in the view, imaging, lighting and occlusion, typical of the real world, as

well as the intra class variations typical of semantic classes of everyday objects.

- *Content-Based Image Retrieval*: The process of retrieving images on the basis of low-level image features, given a query image of a manually constructed description of these low-level features. Such descriptions frequently have little relation to the semantic content of the image.

For the image recognition problem, pattern recognition techniques are applied in various areas such as face recognition [161], character recognition [116], spatial object recognition [129], fingerprint classification [132] and iris recognition [133]. In the recognition of a pattern, a classifier decides whether an input query belong to a specific category or not. As classifiers, linear [54, 164, 76, 177, 116, 57] and nonlinear classifiers [123, 23, 115, 58] has been proposed. In the detail of classifiers, refer to section 3.2.

For the object recognition, as the application of a linear classification method, Murase and Nayar proposed parametric eigenspace method for recognition of spatial object and its pose estimation [129]. Lowe proposed an object recognition method based on local feature and a nonlinear classifier [110]. In this object recognition method, Lowe's feature detection method finds region that are invariant to translation, rotation ans scaling.

After object recognition methods were constructed, object detection methods are developed [105, 41, 46, 182, 45, 16, 165]. Dalal and Triggs [41] proposed an histogram of oriented gradients method, which consists of a local feature extraction method and a classifier method, for pedestrian detection. The histogram of oriented gradients method detects objects with sharp boundaries and a uniform background, such as pedestrians and cars on pavements and streets, respectively, with high accuracy. The extended methods of HoG method are proposed [46, 182, 45, 16]. By using Lowe's local feature and a classifier, Vedaldi *et al.* developed object detection method [165].

In the image categorisation, the bag-of-visual words method [153] is a common approach. In the bag-of-visual words method, transformation-invariant local features, which are extracted from learning images, are used for the generation of a codebook. This codebook is a set of representative extracted local features. The bag-of-visual words method assumes that an image is a set of local features. Therefore, images are represented as histograms that represent the frequencies of occurrence of visual words in the codebook. Csurka *et al.* [40] applied the BoW method for visual categorisation. Sivic and Zisserman [152] applied the BoW method for object and image retrievals.

These applications deal with image patterns, since the purpose of the computer vision is to understand the real three-dimensional world from two-

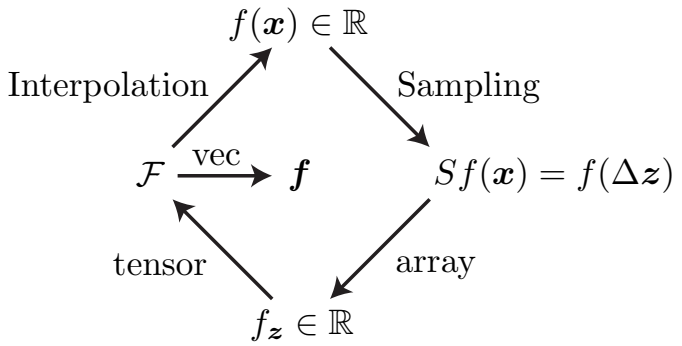


Figure 1.1: Sampling, vectors and tensors. The sampled value $f(\Delta \boldsymbol{z}), \boldsymbol{z} \in \mathbb{Z}^n$ of a function $f(\boldsymbol{x}), \boldsymbol{x} \in \mathbb{R}^n$ yields an array $f_{\boldsymbol{z}}, \boldsymbol{z} \in \mathbb{Z}^{m \times m \times \cdots \times m}$. This array $f_{\boldsymbol{z}}$ is expressed as a tensor $\mathcal{F}$ to preserve multilinearity of $f(\boldsymbol{x})$. Interpolation procedure reconstruct $f(\boldsymbol{x})$ from $Sf(\boldsymbol{x})$ through $\mathcal{F}$. The vector $\boldsymbol{f}$ whose elements are sample values of $f(\boldsymbol{z})$ is constructed from $\mathcal{F}$ by vectorisation operator vec to the tensor $\mathcal{F}$.

dimensional images. To understand the real world, an essential task is to recognise what exists in a two-dimensional image. If we construct a method to recognise objects, we can construct detection method based on the recognition methods. Using the results of a detection procedure, we can understand objects which exist in an image. Once we understand which objects exist in images, we can categorise images with respect to the contents of images. After the categorisation of images, we can construct retrieval methods. Therefore, a robust and accurate recognition method is an essential for computer vision.

For numerical computation of pattern recognition, we deal with sampled patterns. In traditional pattern recognition, these sampled patterns are embedded in an appropriate-dimensional Euclidean space as vectors. The other way is to deal with sampled patterns as higher-dimensional array data. These array data are expressed by tensor to preserve multilinearity of function in the original pattern space. Tensors allow expressing multidimensional array data in multilinear forms. Figure 1.1 illustrates the relation among sampling, vectors and tensor representation for multilinear structure. Furthermore, for tensor space, we can define inner product, norm and distance. These properties allow us to construct pattern recognition method in tensor space.

A pattern is assumed to be a square integrable function defined on a finite support in n-dimensional Euclidean space. For planar and volumetric pattern, the dimensions of Euclidean spaces are two and three, respectively. Organs are essentially spatial textures which are functions defined in three-

dimensional Euclidean space. Furthermore, for video sequence [120] and volumetric sequence [169], the dimensions of the data spaces are three and four, respectively, since in these applications for planar- and spatio-temporal data are focused to analysis. Moreover, planar multichannel images [103, 51] are also expressed as three-way arrays. For these data, elements of two-dimensional array use an additional axis to express frequencies of elements. Multichannel image pattern recognition has been a central issue in remote sensing of earth and planets [26]. In seismic data analysis, the dimension of the data space is five, since waves stated by a planar source array migrated to planar receiver array are focused to analyse [49]. These data, the dimension of whose array are higher than two, can be represented by higher-order tensors.

Moreover, for applications of modern pattern recognition techniques such as deep learning [38] and machine learning for big data [43], we are mathematically and numerically required to evaluate the performance of tensor-based pattern recognition of multilinear data. Importantly, for fast image pattern recognition, a compact representation of these image data is desirable. Tensor expressions fulfill these requirements in applications of pattern recognition of multidimensional array data.

Based on these motivations, we represent multidimensional data by multilinear forms. Using multilinear forms, we construct tensor subspace method that directly operate multidimensional array as multiway data. The operations of tensor have analogies to the operations of vectors. These analogies derive the extension of traditional recognition methods. Therefore, we extend mutual subspace method to multilinear forms. These methods achive fast and accurate recognition of high-dimensional array data for recognition, understand, categorisation and retrieval. Finally, we present numerical experiments of extended methods for three-dimensional spatial data and four-dimensional spatiotemporal data. For the spatial data, we use volumetric data of human organ. For the spatiotemporal data, we use a sequence of volumetric data, which represents beating human heart. The results in numerical experiments show that our recognition methods in multilinear forms are valid for recognition in medical imaging.

1.2 Related Works

Karhunen-Loéve method were independently developed by Karhunen [84] and Loéve [108] at almost the same time. A discrete form of Karhunen-Loéve method was proposed as principal component analysis by Hotelling [73]. For the large scale computation of the principal component analysis, online algorithms are proposed [12, 148, 24]. Iijima [76] and Watanabe [175, 176, 177]

independently introduced the KL expansion for the linear approximation of subspaces of multiple-category data at the almost same time. For practical computation for sampled data, they used the principal component analysis. The principal component analysis selects the subspace in which the covariance of class data is maximised. Watanabe's method was originally called self-featuring information compression (SELFIC) method [175], which based on the minimisation of an entropy in a category. Iijima's method was originally called multiple similarity method. This Iijima's method almost corresponds to Watanabe's class-featuring information compression (CLAFIC) method [176]. Therefore, these methods are called subspace methods. The name subspace method was given by Watanabe *et al.* [176].

As the extension of the subspace method, Iijima [76] introduced the constant normalisation of subspaces to the problem of character recognition. Iijima proposed compound similarity method as the extension of his multiple similarity method. The constant normalisation in the principal component analysis subtracts a constant bias since each image pattern contains a constant bias. Kobayashi *et al.* introduced a cone-restriction to a subspace of category, and proposed cone-restricted subspace method [90] . This cone-restriction assumes that feature vectors are occasionally subject to non-negative constraints in pattern recognition. This non-negative constraints are represented by a cone in feature space. As described in ref. [116], the variants of subspace method have been studied by many researchers, such as Fu and Yu [56], Kittler and Young [88], and Kohonen [92], and so on. For further details of the variants and the history of the subspace method, refer to Oja's[130] book and Grenander's book [63], respectively.

Maeda [117, 116] proposed the mutual subspace method that computes the orthogonal projection of subspaces spanned by inputs with perturbations. Fukui and his co-workers [59, 57] proposed a combination of a generalisation of constant normalisation and the mutual subspace method. This method subtracts the elements in the common linear subspace of many categories. In the mutual subspace method, classification is based on angles among subspaces of each category. Cock and Moor [36] introduced the notion of subspace angles by considering the principal angles between two subspaces. Hamm and Lee [65] proposed a framework for Grassmann manifold learning. A Grassmann manifold is the set of fixed-dimensional linear subspaces in a Euclidean space. This learning framework unifies the view of the subspace-based learning method by formulating problems on a Grassmann manifold. The Grassmann manifold manipulate each subspace as a point in the Grassmann space, and feature extraction and classification are performed in the same space. Krim and his co-workers [17, 151] focus on recovering of union of subspaces. To discover the underlying structure of high-dimensional data,

they built on the model of union of subspaces. In ref. [151], they adopt the fundamental structure of a Grassmann manifold and proposed a technique of pursuit the subspace.

Kernel methods are nonlinear methods. For the nonlinear principal component analysis, that is kernel principal component analysis, Schölkopf *et al.* [149] introduced kernel trick that computes inner norm of two data in high-dimensional space without mapping original two data into a high-dimensional space. These projected data can be separated by hyperplanes in high-dimensional space, although they cannot be separated by hyperplanes in the original space. Therefore, in this high-dimensional space, we can use the principal component analysis as kernel principal component analysis.

Applying kernel trick to subspace method, Maeda and Murase has extended the subspace method to kernel subspace method subspace [115]. They claimed that kernel subspace method achieved more accurate recognition than the traditional subspace method, since traditional subspace methods fail when the pattern distribution has nonlinear characteristics or the feature space dimension is low compared to the number of classes by the experimental results for phantom data and hand-printed Japanese katakana characters. Sakano and his co-workers also applied kernel trick to the mutual subspace method, and proposed kernel mutual subspace for face and object recognition [144, 145]. Furthermore, Fukui *et al.* integrated kernel trick and generalised constant normalisation for the mutual subspace method as kernel constrained mutual subspace method [58] for spatial object recognition. Moreover, Kobayashi *et al.* integrated kernel trick and cone-restriction for the subspace method as cone-restricted kernel subspace method [91].

The above methods are constructed for without multilinear properties in multiway data. Therefore, we need to construct a classification method in a multilinear form for multiway data.

For the dimension reduction of multilinear patterns, principal component analysis was extended to multilinear forms. The origin of the tensor principal component analysis for the third-order tensors was proposed as the decomposition of tensors by Tucker [160]. For the Tucker decomposition of second- and third-order tensors, Kroonenberg and Jeeuw discussed the properties of convergence of alternating-least-squares algorithms [95]. In general for Tucker decomposition, orthogonality constraints on decomposed tensors are not required. Cichoki *et al.* imposed that the existence of the constraints is the difference between the tensor principal component analysis and parallel factor analysis [35]. In ref. [35], in addition to orthogonal constraints, sparse constraints and nonnegative constraints for tensor decomposition are studied.

Second-order tensor principal component analysis, which directly decom-

poses an image matrix, is used for two-dimensional images [111] as an extension of the principal component analysis. According to a review on multilinear subspace learning [111], there are three basic projections for a tensor. The second-order tensor principal component analysis uses a tensor-to-tensor projection consisting of 1- and 2-mode projections that act on columns and rows of images, respectively. Yang *et al.* [184] proposed two-dimensional principal component analysis for image representation. Otsu [131] developed the marginal eigenvector method, which is based on both 1- and 2-mode projections. Aase *et al.* [1] developed a singular value decomposition-based image coding system. Ding and Ye [44] developed the two-dimensional singular value decomposition, which is equivalent to the coding system of Aase *et al.*, as an extension of the singular value decomposition for image compression. The two-dimensional singular value decomposition is also based on both 1- and 2-mode projections. The projections in the marginal eigenvector method and two-dimensional singular value decomposition are equivalent to the tensor-to-tensor projection for a second-order tensor. This mathematical property implies that the two-dimensional two-dimensional singular value decomposition is a special case of the tensor principal component analysis.

Ye *et al.* [185] proposed generalised principal component analysis for image compression that finds both 1- and 2-mode projections. The generalised principal component analysis is a two-dimensional version of the iterative algorithm for the SVD [125]. Furthermore, the generalised principal component analysis is a second-order version of the multilinear principal component analysis [112], which is an practical computation method of the tensor principal component analysis. Moreover, in the multilinear principal component analysis, the projections obtained by higher-order singular value decomposition [100] is used as the initial projections of the iterative algorithm. The iterative algorithms in the generalised principal component analysis, multilinear principal component analysis and higher-order singular value decomposition are called alternating least squares algorithm. Independently from the multilinear principal component analysis, tensor rank-one decomposition [172] was proposed for multidimensional data compression. The tensor rank-one decomposition is also an iterative algorithm based on the alternating least squares algorithm and the higher-order singular value decomposition. The minimisation problem in the tensor rank-one decomposition is coincident with that of the multilinear principal component analysis. A difference between the multilinear principal component analysis and tensor rank-one decomposition is that the tensor rank-one decomposition finds rank-one tensors as bases for a tensor subspace, while the multilinear principal component analysis finds bases for each mode of tensors. As the extensions of the multilinear principal component analysis, by adding uncorrelation and sparsity

constraints to the minimisation problem of decomposition in the multilinear principal component analysis, uncorrelated multilinear principal component analysis [113] and sparse higher-order principal component analysis [8] were proposed, respectively.

The extended methods of multilinear principal component analysis have been developed [77, 113, 8]. Robust multilinear principal component analysis [77] is a robust version of tensor principal component analysis for image pattern recognition including outliers. Uncorrelated multilinear principal component analysis [113] searches for a tensor-to-vector projection that obtains most of the variation in the original tensorial input by deciding the maximum number of uncorrelated features. Sparse higher-order principal component analysis [8] searches for the minimum number of bases for input tensors by assuming sparsity in tensor decomposition.

Using the tensor principal component analysis, we construct recognition methods in multilinear forms for multiway data of multi categories.

1.3 Organization of the Dissertation

Figure 1.2 illustrates the relations among chapters in this dissertation. In Chapter 2, we introduce tensor representations and computational methods in multilinear forms. We define tensors and its multiway access methods, that is unfolding, vectorising and n-mode product, for first-, second-, third- and Nth-order. Then, using these introduced multilinear forms, we present principal component analysis and discrete cosine transform for first-, second-, third- and Nth-order tensors.

In Chapter 3, focusing two-dimensional images, we mathematically and experimentally show the effects of dimension-reduction methods for pattern recognition in linear and bilinear forms. This chapter summarises topological and geometrical properties that required for the image pattern recognition. As dimension reduction methods, we adopt naive downsampling, pyramid transform, two-dimensional discrete cosine transform, random projection, two-dimensional random projection, marginal eigenvalue, multidimensional scaling. As linear classifier, we adopt subspace method, mutual subspace method, constraint mutual subspace method. As bilinear classifier, we adopt two-dimensional tensor subspace method.

In Chapter 4, we propose two recognition methods in multilinear form for higher-order tensors: tensor subspace method and mutual tensor subspace method for Nth-order tensors. In experiments, we validate recognition accuracy of the tensor subspace method for gait data in computer vision and volumetric data in medical imaging. Furthermore, we validate recognition

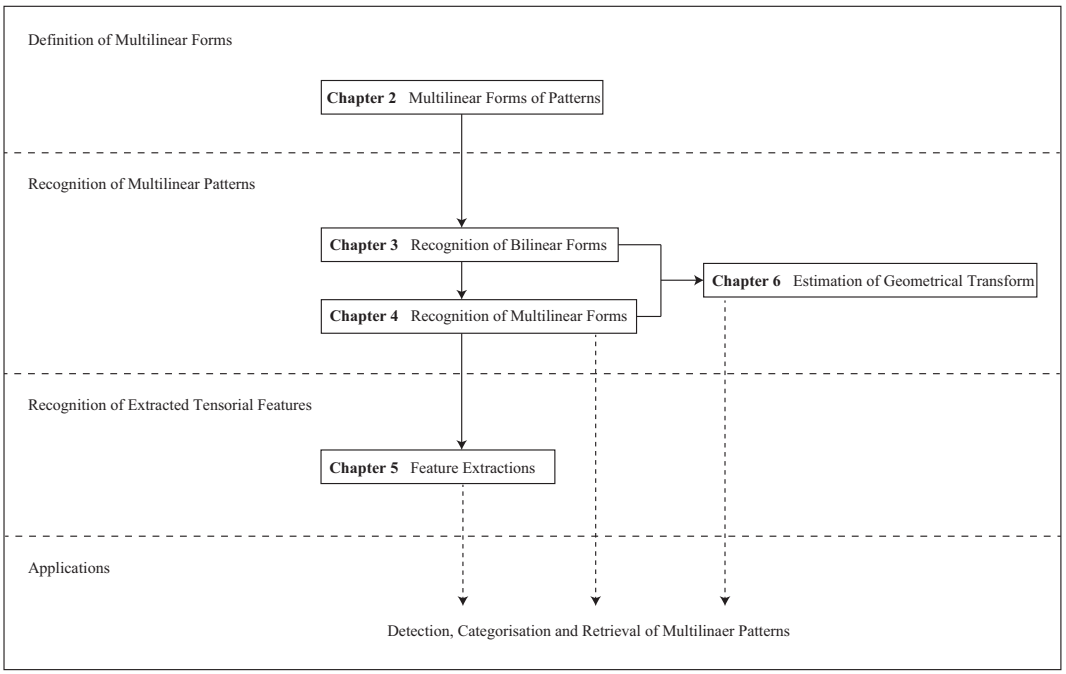


Figure 1.2: The relations among chapters in this dissertation.

accuracy of mutual tensor subspace method for sequences of volumetric data of beating human heart by comparing with the results of tensor subspace method.

In Chapter 5, we explore essential feature in gradient filed in two-dimensional images. Gradient field on an image is represented by third-order tensors. In the object recognition and detection methods, this gradient field are partitioned to small regions and represented by a set of histograms of orientation of gradients in these small regions. For these histograms, we introduce L_p-norms for $p = 1, 2$ and Wasserstein distance. Using these metric and several partitioning methods, we validate the accuracy of image recognition for two-category: pedestrian and background.

In Chapter 6, we propose local linear method for dictionary-based image registration of two- and three-dimensional images. This method assumes the local linearity on an image manifold. Using local subspace around a point on image manifold, for template image with geometrical perturbation, we can generate new entry for pre-stored images in dictionary, and estimate geometrical transform between input and pre-stored images.

Chapter 2

Multilinear Forms ofPattern

This Chapter is partly based on Publications of Journal Papers "1. Pattern Recognition in Multilinear Space and its Applications: Mathematics, Computational Algorithms and Numerical Validations" and "2. Dimension Reduction and Construction of Feature Space for Image Pattern Recognition", and Publication of International Conference "1, Approximation of N-Way Principal Component Analysis for Organ Data".

2.1 Multilinear Form

2.1.1 Preliminaries

Let $\mathcal{K}$ to be a scalar field. $\mathcal{K}$ is a set of numbers that is closed with respect to addition and multiplication. For $a, b, c \in \mathcal{K}$, the following conditions hold:

- commutativity: $a + b = b + c \in \mathcal{K} \quad ab = ba \in \mathcal{K}$;
- associativity: $(a + b) + c = a + (b + c) \in \mathcal{K} \quad (ab)c = a(bc) \in \mathcal{K}$;
- distritutivity of multiplication over addition: $a(b+c) = (ab)+(ac) \in \mathcal{K}$;
- existence of identity element: $0 + a = a, 0 \in \mathcal{K} \quad 1a = a, 1 \in \mathcal{K}$;
- existence of inverse of addition : $\forall a \in \mathcal{K}$, $\exists a' \in \mathcal{K}$ such that $a + a' = 0$; $\forall a \in \mathcal{K} \setminus 0$, $\exists a' \in \mathcal{K}$ such that $aa' = 1$.

The set $\mathbb{R}$ of the real numbers with addition and multiplication is also a field. In this dissertation, we use this set of real numbers. A linear space $\mathcal{V}$, that is a vector space, is defined over a filed $\mathcal{K}$. For scalars $a, b \in \mathcal{K}$ and vectors $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathcal{V}$, the following conditions hold:

- $\mathcal{V}$ is closed with respect to addition of vectors,
  - Associativity: $(\boldsymbol{x}+\boldsymbol{y})+\boldsymbol{z}=\boldsymbol{x}+(\boldsymbol{y}+\boldsymbol{z})\in\mathcal{V}$;
  - commutativity: $\boldsymbol{x}+\boldsymbol{y}=\boldsymbol{y}+\boldsymbol{x}\in\mathcal{V}$;
  - existence of identity element: $\mathbf{0}+\boldsymbol{x}=\boldsymbol{x},\mathbf{0}\in\mathcal{V}$;
  - existence of inverse element: $\forall\boldsymbol{x}\in\mathcal{V}$, $\exists\boldsymbol{x}'\in\mathcal{V}$ such that $\boldsymbol{x}+\boldsymbol{x}'=\mathbf{0}$;
- $\mathcal{V}$ is closed with respect to multiplication of vector by a scalar,
  - distributivity: $(a+b)\boldsymbol{x}=a\boldsymbol{x}+b\boldsymbol{y}\in\mathcal{V}$;
  - distributivity: $a(\boldsymbol{x}+\boldsymbol{y})=a\boldsymbol{x}+a\boldsymbol{y}\in\mathcal{V}$;
  - associativity: $(ab)\boldsymbol{x}=a(b\boldsymbol{x})\in\mathcal{V}$;
  - existence of identity element: $1\boldsymbol{x}=\boldsymbol{x}$.

A map from a linear space $\mathcal{V}$ to another linear space $\mathcal{V}'$, we have a following theorem.

Theorem 2.1 *For a scalar $a\in\mathcal{K}$ and vectors $\boldsymbol{x},\boldsymbol{y}\in\mathcal{V}$, a map T from a linear space $\mathcal{V}$ to another linear space $\mathcal{V}'$ is a linear map if*

$$T(\boldsymbol{x}+\boldsymbol{y})=T(\boldsymbol{x})+T(\boldsymbol{y}), \tag{2.1}$$

$$T(a\boldsymbol{x})=aT(\boldsymbol{x}). \tag{2.2}$$

In a linear space, a vector is defined as an one-dimensional array that contains elements of $\mathcal{K}$. By extending a one-dimension array to a higher-dimensional array, we have a tensor.

A tensor space $\mathcal{T}$ is also defined over a filed $\mathcal{K}$. For scalars $a,b\in\mathcal{K}$ and tensors $\mathcal{X},\mathcal{Y},\mathcal{Z}\in\mathcal{T}$, the following conditions hold:

- $\mathcal{T}$ is closed with respect to addition of tensors,
  - Associativity: $(\mathcal{X}+\mathcal{Y})+\mathcal{Z}=\mathcal{X}+(\mathcal{Y}+\mathcal{Z})\in\mathcal{T}$;
  - commutativity: $\mathcal{X}+\mathcal{Y}=\mathcal{Y}+\mathcal{X}\in\mathcal{T}$;
  - existence of identity element: $\boldsymbol{O}+\mathcal{X}=\mathcal{X},\boldsymbol{O}\in\mathcal{T}$;
  - existence of inverse element: $\forall\mathcal{X}\in\mathcal{T}$, $\exists\mathcal{X}'\in\mathcal{T}$ such that $\mathcal{X}+\mathcal{X}'=\boldsymbol{O}$;
- $\mathcal{T}$ is closed with respect to multiplication of vector by a scalar,
  - distributivity: $(a+b)\mathcal{X}=a\mathcal{X}+b\mathcal{Y}\in\mathcal{T}$;

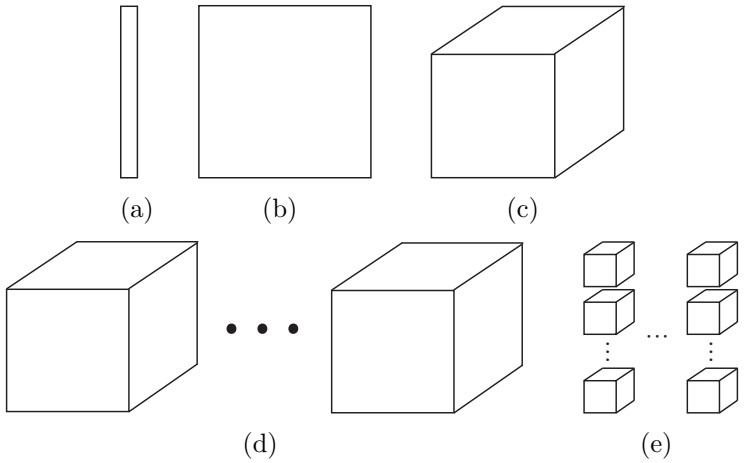


Figure 2.1: Examples of tensors. (a) a first-order tensor, that is a vector. (b) a second-order tensor, that is a matrix. (c) a third-order tensor. (d) a fourth-order tensor. (3) an Nth-order tensor, which is a sequence of $(N-1)$th-order tensors.

- distributivity: $a(\mathcal{X} + \mathcal{Y}) = a\mathcal{X} + a\mathcal{Y} \in \mathcal{T}$;
- associativity: $(ab)\mathcal{X} = a(b\mathcal{X}) \in \mathcal{T}$;
- existence of identity element: $1\mathcal{X} = \mathcal{X}$.

Therefore, a tensor space is also linear space. However, a tensor space has a different property from a linear space of vectors, since a tensor is a multidimensional array.

The dimension of array for a tensor is specified by term order. A scalar is a zero-order tensor. One-dimensional and two-dimensional array, that is vector and matrix, are second-order and third-order tensors, respectively. For Nth-order tensor, the entries of a tensor are addressed by N indexes. Figure 2.1 illustrates examples of first-, second, third-, fourth and Nth-order tensors. Each index defined one *mode*. By using unfolding operator, we have a set of vectors with respect to one mode. For this unfolded tensor, we can apply linear transform with respect to one-dimensional array of only one mode of Nth-order tensor. Therefore, a linear map of a tensor is written as successive linear maps by

$$T(\mathcal{X}) = T_N(T_{N-1}(\dots T_2(T_1(\mathcal{X})))) \tag{2.3}$$

where and T_i for $i = 1, 2, \dots, N$ are linear maps with respect to i-mode vectors of a tensor $\mathcal{X}$. This is a multilinear map.

For a tensor, a multilinear projection maps the input tensor data from one space to another space. We have three basic multilinear projections, that is, the vector-to-vector projection, tensor-to-vector projection and tensor-to-tensor projection (TTP). The vector-to-vector projection is a linear projection from a vector to another vector. To use the vector-to-vector projection for tensors, we need to reshape tensors into vectors before the projection. The tensor-to-vector projection, which is also referred to as the rank-one projection [173, 156, 75], consists of elementary multilinear projections. An elementary multilinear projection projects a tensor to a scalar. Using d elementary multilinear projections, the tensor-to-vector projection obtains a d-dimensional vector projected from a tensor. The TTP projects a tensor to another tensor of the same order. In this paper, we focus on methods of finding the optimal projection for the TTP.

In the following subsections, we briefly summarise the multilinear projection for first-, second-, third- and Nth-order tensors.

2.1.2 First-Order Tensor

For a vector $\boldsymbol{x} = ((x_i)) \in \mathbb{R}^{I_1}$ and a integer $P_1 \leq I_1$, we set orthogonal matrix $\boldsymbol{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{I_1})$, where $\boldsymbol{u}_i \in \mathbb{R}^{I_1}$ for $i = 1, 2, \ldots, P_1$ are linear independent vectors. Using this orthogonal matrix, we have a linear projection

$$\boldsymbol{y} = \boldsymbol{U}^\top \boldsymbol{x}. \tag{2.4}$$

We define the inner product of two vectors $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^{I_1}$ by

$$\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle = \sum_{i_1}^{I_1} x_{1,i} x_{2,i}. \tag{2.5}$$

Using this inner product, the Euclidean norm of a vector $\boldsymbol{x}$ is

$$\|\boldsymbol{x}\|_2 = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}. \tag{2.6}$$

For the two vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, we define the distance between them as

$$d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2. \tag{2.7}$$

2.1.3 Second-Order Tensor

For a matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$, setting orthogonal matrices $\boldsymbol{U}^{(1)} \in \mathbb{R}^{m \times m}$ and $\boldsymbol{U}^{(2)} \in \mathbb{R}^{n \times n}$, we have a bilinear projection

$$\boldsymbol{Y} = \boldsymbol{U}^{(1)\top} \boldsymbol{X} \boldsymbol{U}^{(2)} \tag{2.8}$$

and its inverse transform

$$\boldsymbol{X} = \boldsymbol{U}^{(1)}\boldsymbol{Y}\boldsymbol{U}^{(2)\top}. \tag{2.9}$$

In this transform, left and right operators act on column and row vectors of $\boldsymbol{X}$, respectively. This bilinear projection is special case of the TTP.

Replacing $I_1 = m$ and $I_2 = n$, we define a second-order tensor

$$\mathcal{X} = ((x_{ij})) \tag{2.10}$$

with a pair of indeces $1 \leq i \leq I_1$ and $1 \leq j \leq I_2$, which is the matrix $\boldsymbol{X} = ((x_{i_1 i_2})) \in \mathbb{R}^{I_1 \times I_2}$. Indeces i and j denote the index for 1- and 2-modes of a tensor $\mathcal{X}$. For the outer products of two vectors $\boldsymbol{u}^{(1)}$ and $\boldsymbol{u}^{(2)}$, if a tensor $\mathcal{X}$ satisfies the condition

$$\mathcal{X} = \boldsymbol{u}^{(1)} \circ \boldsymbol{u}^{(2)}, \tag{2.11}$$

where $\circ$ denotes the outer product, we call this second-order tensor a rank-one tensor. For a tensor $\mathcal{X}$, the unfolding of $\mathcal{X}$ is defined by

$$\mathcal{X}_{(1)} = \boldsymbol{X} \in \mathbb{R}^{I_1 \times I_2}, \quad \mathcal{X}_{(2)} = \boldsymbol{X}^\top \in \mathbb{R}^{I_2 \times I_1}. \tag{2.12}$$

Figure 2.2 shows unfoldings for 1- and 2- modes. For a tensor $\mathcal{X} = \mathcal{X}_{(1)} = \boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{I_2})$, where $\boldsymbol{x}_i \in \mathbb{R}^{I_1}$ for $i = 1, 2, \ldots, I_2$, we define vectorising operator a for second-order tensor by

$$\text{vec } \mathcal{X} = (\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top, \ldots, \boldsymbol{x}_{I_2}^\top)^\top. \tag{2.13}$$

The 1- and 2-mode products of a tensor by matrices $\boldsymbol{U}^{(1)\top} \in \mathbb{R}^{I_1 \times I_1}$ and $\boldsymbol{U}^{(2)\top} \in \mathbb{R}^{I_2 \times I_2}$ are, respectively, given by

$$\mathcal{X} \times_1 \boldsymbol{U}^\top = \boldsymbol{U}^\top \boldsymbol{X}_{(1)} = \boldsymbol{U}^\top \boldsymbol{X}, \tag{2.14}$$

$$\mathcal{X} \times_2 \boldsymbol{U}^\top = \boldsymbol{U}^\top \boldsymbol{X}_{(2)} = \boldsymbol{U}^\top \boldsymbol{X}^\top, \tag{2.15}$$

respectively.

As the tensor $\mathcal{X}$ is in the tensor space $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2}$, the tensor space can be interpreted as the Kronecker product of three vector spaces $\mathbb{R}^{I_1}, \mathbb{R}^{I_2}$. To project $\mathcal{X} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2}$ to another tensor $\mathcal{Y}$ in a lower-dimensional tensor space $\mathbb{R}^{P_1} \otimes \mathbb{R}^{P_2}$, where $P_n \leq I_n$ for $n = 1, 2$, we need three matrices $\{\boldsymbol{U}^{(n)} \in \mathbb{R}^{I_n \times P_n}\}_{n=1}^2$. Using two matrices, we have the TTP

$$\mathcal{Y} = \mathcal{X} \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} = \boldsymbol{U}^{(1)\top} \boldsymbol{X} \boldsymbol{U}^{(2)}, \tag{2.16}$$

which projects $\mathcal{X}$ to a lower-dimensional tensor space. This TTP is the tensor representation of the bilinear projection in eq. (2.8). Figure 2.4 shows the TTP for a second-order tensor.

We define the inner product of two tensors $\mathcal{X}_1, \mathcal{X}_2 \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ by

$$\langle \mathcal{X}_1, \mathcal{X}_2 \rangle = \sum_{i_1}^{I_1} \sum_{i_2}^{I_2} \mathcal{X}_1(i_1, i_2) \cdot \mathcal{X}_2(i_1, i_2). \tag{2.17}$$

Using this inner product, the Frobenius norm of a tensor $\mathcal{X}$ is

$$\|\mathcal{X}\|_{\mathrm{F}} = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}. \tag{2.18}$$

For the Frobenius norm of a tensor, we have

$$\|\mathcal{X}\|_{\mathrm{F}} = \|\mathrm{vec}\ \mathcal{X}\|_2, \tag{2.19}$$

where $\|\cdot\|_2$ is Euclidean norm of a vector. For the two tensors $\mathcal{X}_1$ and $\mathcal{X}_2$, we define the distance between them as

$$d(\mathcal{X}_1, \mathcal{X}_2) = \|\mathcal{X}_1 - \mathcal{X}_2\|_{\mathrm{F}}. \tag{2.20}$$

Although this definition is a tensor-based measure, this distance is equivalent to the Euclidean distance between the vectorised tensors $\mathcal{X}_1$ and $\mathcal{X}_2$.

2.1.4 Third-Order Tensor

A tensor $\mathcal{M} \in \mathbb{R}^{m \times n}$, that is a matrix, is expressed as $((x_{ij}))$ for $1 \leq i \leq I_1$, $1 \leq j \leq I_2$. Therefore, as the extension of an matrix, a third-order tensor is defined in $\mathbb{R}^{I_1 \times I_2 \times I_3}$. A third-order tensors is expressed as

$$\mathcal{X} = ((x_{ijk})) \tag{2.21}$$

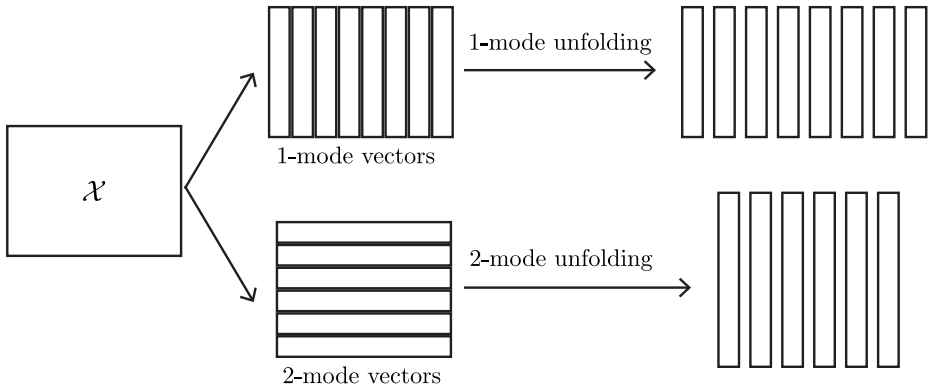
with three indeces $1 \leq i \leq I_1$, $1 \leq j \leq I_2$, $1 \leq k \leq I_3$. i, j, k denote the mode of a tensor $\mathcal{X}$. For the outer products of three vectors $\boldsymbol{u}^{(1)}$, $\boldsymbol{u}^{(2)}$ and $\boldsymbol{u}^{(3)}$, if the tensor $\mathcal{X}$ satisfies the condition

$$\mathcal{X} = \boldsymbol{u}^{(1)} \circ \boldsymbol{u}^{(2)} \circ \boldsymbol{u}^{(3)}, \tag{2.22}$$

where $\circ$ denotes the outer product, we call this tensor $\mathcal{X}$ a rank-one tensor. For $\mathcal{X}$, the n-mode vectors, $n = 1, 2, 3$, are defined as the I_n-dimensional vectors obtained from $\mathcal{X}$ by varying index i_n while fixing all the other indices. For $n = 1, 2, 3$ the unfolding of $\mathcal{X}$ along the n-mode vectors of $\mathcal{X}$ is defined as

$$\mathcal{X}_{(1)} \in \mathbb{R}^{I_1 \times I_{23}},\ \mathcal{X}_{(2)} \in \mathbb{R}^{I_2 \times I_{13}},\ \mathcal{X}_{(3)} \in \mathbb{R}^{I_3 \times I_{12}} \tag{2.23}$$

where $I_{23} = I_2 \times I_3$, $I_{13} = I_1 \times I_3$, $I_{12} = I_1 \times I_2$, and the column vectors of $\mathcal{X}_{(n)}$ are the n-mode vectors of $\mathcal{X}$. Figure 2.5 shows an example of n-mode



(a)

Figure 2.2: 1- and 2-mode unfoldings of a second-order tensor $\mathcal{X} \in \mathbb{R}^{6\times 8}$.

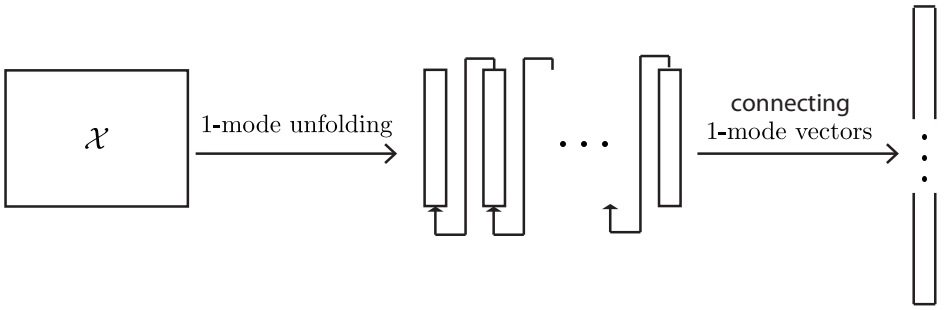
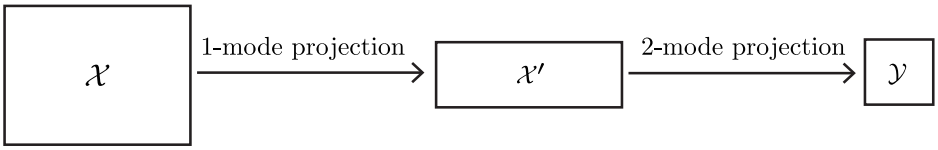
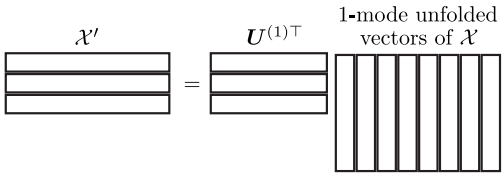


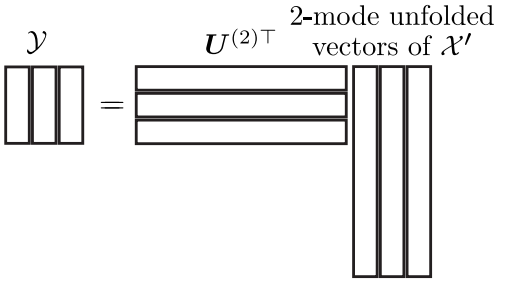
Figure 2.3: Vectorising of a second-order tensor. By connecting unfolded 1-mode vectors, we have a long vector.



(a) Tensor-to-tensor projection for a tensor $\mathcal{X}$



(b) 1-mode projection for $\mathcal{X}$ represented by a linear projection



(c) 2-mode projection for $\mathcal{X}$ represented by a linear projection

Figure 2.4: Tensor-tensor projection of a second-order tensor $\mathcal{X} \in \mathbb{R}^{6\times 8}$ to a lower-dimensional tensor $\mathcal{Y} \in \mathbb{R}^3$.

unfolding for a third-order tensor. For a tensor $\mathcal{X}$, using vectorising operator for a second-order tensor and unfolding of a third-order tensor, we define vectorising operator for a third-order tensor by

$$\text{vec } \mathcal{X} = \text{vec } \mathcal{X}_{(1)}, \tag{2.24}$$

where operators in left and right handsites are vectorising operators for third-order and a second-order tensors, respectively.

The 1-, 2- and 3-mode products of a matrices $\boldsymbol{U}^{(1)} \in \mathbb{R}^{I_1 \times P_1}$, $\boldsymbol{U}^{(2)} \in \mathbb{R}^{I_2 \times P_2}$ and $\boldsymbol{U}^{(3)} \in \mathbb{R}^{I_3 \times P_3}$ and a tensor $\mathcal{X}$ are given by

$$\mathcal{X} \times_1 \boldsymbol{U}^{(1)\top} = \hat{\mathcal{X}}^{(1)},\ \hat{\mathcal{X}}^{(1)}_{(1)} = \boldsymbol{U}^{(1)\top} \mathcal{X}_{(1)}, \tag{2.25}$$

$$\mathcal{X} \times_2 \boldsymbol{U}^{(2)\top} = \hat{\mathcal{X}}^{(2)},\ \hat{\mathcal{X}}^{(2)}_{(2)} = \boldsymbol{U}^{(2)\top} \mathcal{X}_{(2)}, \tag{2.26}$$

$$\mathcal{X} \times_3 \boldsymbol{U}^{(3)\top} = \hat{\mathcal{X}}^{(3)},\ \hat{\mathcal{X}}^{(3)}_{(3)} = \boldsymbol{U}^{(3)\top} \mathcal{X}_{(3)}, \tag{2.27}$$

where $\hat{\mathcal{X}}^{(1)}_{(1)}$, $\hat{\mathcal{X}}^{(2)}_{(2)}$ and $\hat{\mathcal{X}}^{(3)}_{(3)}$ are unfolded tensors of $\hat{\mathcal{X}}^{(1)}$, $\hat{\mathcal{X}}^{(2)}$ and $\hat{\mathcal{X}}^{(3)}$, respectively. Therefore, n-mode product of $\mathcal{X}$ are achieved by matricising of a tensor, product with a matrix and tensors of the results of the product to a tensor. For two matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, n-mode and m-mode tensor products are commutative [35], that is,

$$\mathcal{X} \times_n \boldsymbol{U} \times_m \boldsymbol{V} = \mathcal{X} \times_m \boldsymbol{V} \times_n \boldsymbol{U}. \tag{2.28}$$

As the tensor $\mathcal{X}$ is in the tensor space $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \mathbb{R}^{I_3}$, the tensor space can be interpreted as the Kronecker product of three vector spaces $\mathbb{R}^{I_1}, \mathbb{R}^{I_2}, \mathbb{R}^{I_3}$. To project $\mathcal{X} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \mathbb{R}^{I_3}$ to another tensor $\mathcal{Y}$ in a lower-dimensional tensor space $\mathbb{R}^{P_1} \otimes \mathbb{R}^{P_2} \otimes \mathbb{R}^{P_3}$, where $P_n \leq I_n$ for $n = 1, 2, 3$, we need three matrices $\{\boldsymbol{U}^{(n)} \in \mathbb{R}^{I_n \times P_n}\}_{n=1}^{3}$. Using the three matrices, the TTP is given by

$$\mathcal{Y} = \mathcal{X} \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \times_3 \boldsymbol{U}^{(3)\top}. \tag{2.29}$$

This projection is established in three steps, where at the nth step, each n-mode vector is projected to a P_n-dimensional space by $\boldsymbol{U}^{(n)}$. Figure 2.7(a) shows the steps for the projection of a third-order tensor to a lower-dimensional tensor. Figures 2.7(b)-(d) show the procedures used to project third-order tensors.

We define the inner product of two tensors $\mathcal{X}_1, \mathcal{X}_2 \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ by

$$\langle \mathcal{X}_1, \mathcal{X}_2 \rangle = \sum_{i_1}^{I_1} \sum_{i_2}^{I_2} \sum_{i_3}^{I_3} \mathcal{X}_1(i_1, i_2, i_3) \cdot \mathcal{X}_2(i_1, i_2, i_3). \tag{2.30}$$

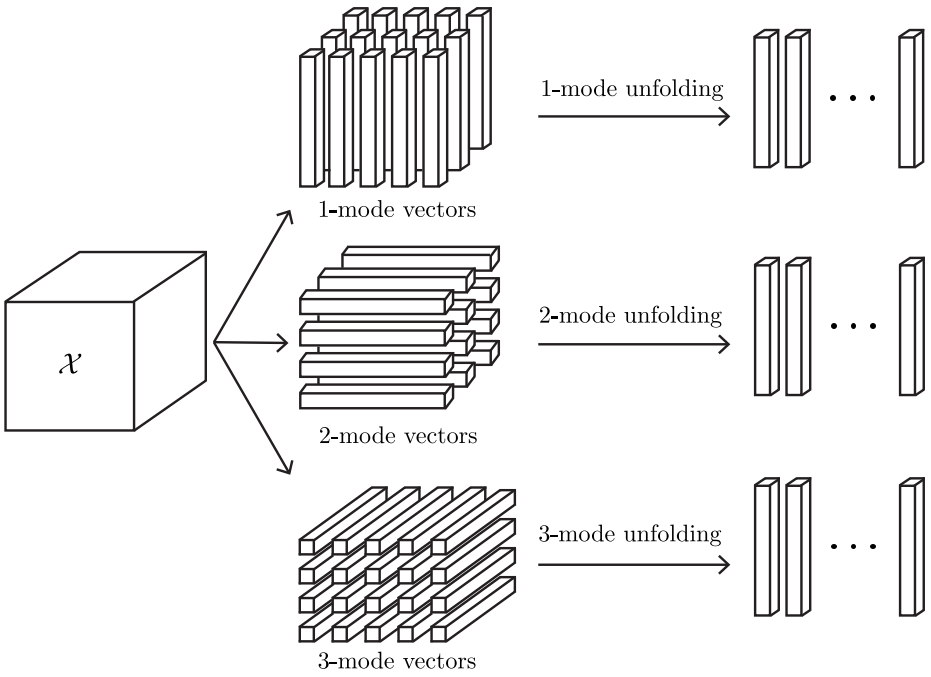


Figure 2.5: Unfoldings of a third-order tensor showing 1-, 2- and 3-mode unfoldings of the third-order tensor $\mathcal{X} \in \mathbb{R}^{4\times5\times3}$.

Using this inner product, the Frobenius norm of a tensor $\mathcal{X}$ is

$$\|\mathcal{X}\|_{\mathrm{F}} = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}. \tag{2.31}$$

For the Frobenius norm of a tensor, we have

$$\|\mathcal{X}\|_{\mathrm{F}} = \|\mathrm{vec}\ \mathcal{X}\|_2, \tag{2.32}$$

where $\|\cdot\|_2$ is Euclidean norm of a vector, respectively. For the two tensors $\mathcal{X}_1$ and $\mathcal{X}_2$, we define the distance between them as

$$d(\mathcal{X}_1, \mathcal{X}_2) = \|\mathcal{X}_1 - \mathcal{X}_2\|_{\mathrm{F}}. \tag{2.33}$$

Although this definition is a tensor-based measure, this distance is equivalent to the Euclidean distance between the vectorised tensors $\mathcal{X}_1$ and $\mathcal{X}_2$.

2.1.5 Nth-Order Tensor

A Nth-order tensor $\mathcal{X}$ defined in $\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ is expressed as

$$\mathcal{X} = (x_{i_1,i_2,\ldots,i_N}) \tag{2.34}$$

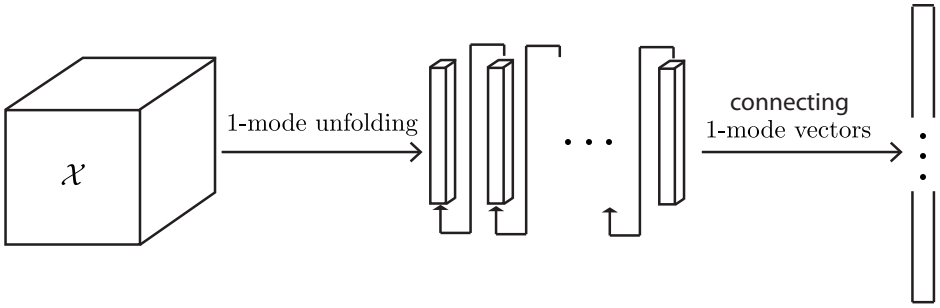


Figure 2.6: Vectorising of a third-order tensor. By connecting unfolded 1-mode vectors, we have a long vector.

for $x_{i_1,i_2,\ldots,i_N} \in \mathbb{R}$, using N indices i_n. Each subscript n denotes the n-mode of $\mathcal{X}$. For $\mathcal{X}$, the n-mode vectors, $n = 1, 2, \ldots, N$, are defined as the I_n-dimensional vectors obtained from $\mathcal{X}$ by varying this index i_n while fixing all the other indices. The unfolding of $\mathcal{X}$ along the n-mode vectors of $\mathcal{X}$ is defined as

$$\mathcal{X}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times I_2 \times \ldots I_{n-1} \times I_{n+1} \times \cdots \times I_N)}, \tag{2.35}$$

where the column vectors of $\mathcal{X}_{(n)}$ are the n-mode vectors of $\mathcal{X}$. For a Nth-order tensor, using vectorising operator for a second-order tensor and unfolding of a Nth-order tensor, we define vectorising operator for a Nth-order tensor by

$$\text{vec } \mathcal{X} = \text{vec } \mathcal{X}_{(1)}, \tag{2.36}$$

where operators in left and right handsite are operators for Nth-order and a second-order tensors, respectively.

The n-mode product $\mathcal{X} \times_n \boldsymbol{U}$ of a matrix $\boldsymbol{U} \in \mathbb{R}^{J_n \times I_n}$ and a tensor $\mathcal{X}$ is a tensor $\mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N}$, with elements

$$g_{i_1,i_2,\ldots,i_{n-1},j_n,i_{n+1},\ldots,i_N} = \sum_{i_n=1}^{I_n} x_{i_1,i_2,\ldots,I_N} u_{j_n,i_n}, \tag{2.37}$$

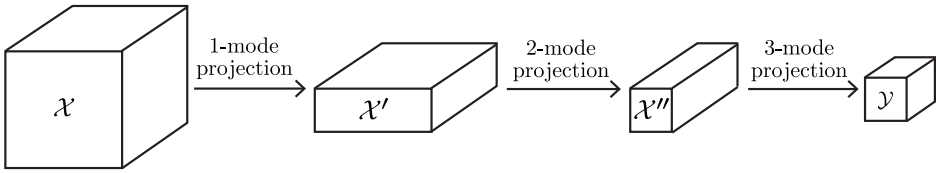
by the manner in ref. [35]. A linear projection form of n-mode product in eq. (2.37) is given by

$$\mathcal{G}_{(n)} = \boldsymbol{U}\mathcal{X}_{(n)}. \tag{2.38}$$

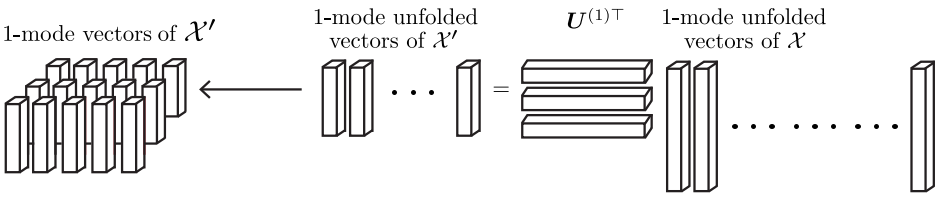
For the m- and n-mode product by matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, we have

$$\mathcal{X} \times_m \boldsymbol{U} \times_n \boldsymbol{V} = \mathcal{X} \times_n \boldsymbol{V} \times_m \boldsymbol{U} \tag{2.39}$$

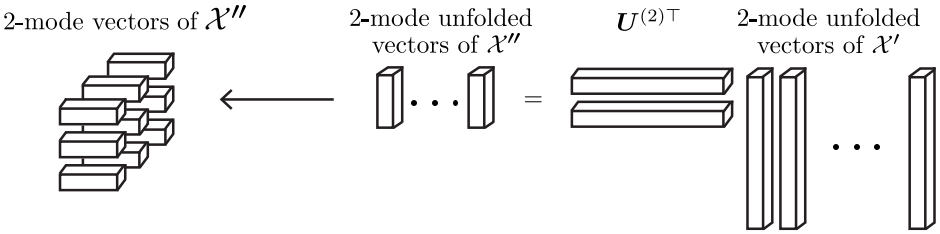
since n-mode projections are commutative [35].



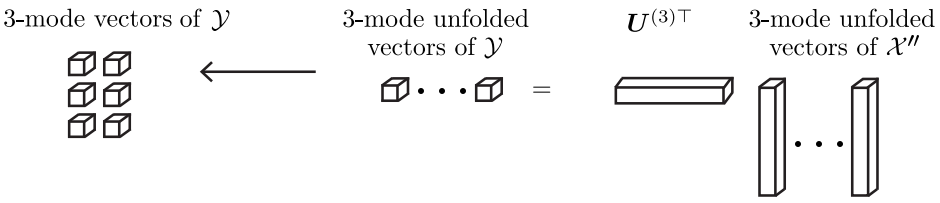
(a) Tensor-to-tensor projection for a tensor $\mathcal{X}$



(b) 1-mode projection for $\mathcal{X}$ represented by a linear projection



(c) 2-mode projection for $\mathcal{X}$ represented by a linear projection



(d) 3-mode projection for $\mathcal{X}$ represented by a linear projection

Figure 2.7: Tensor-tensor projection of a third-order tensor $\mathcal{X} \in \mathbb{R}^{4\times5\times3}$ to a lower-dimensional tensor $\mathcal{Y} \in \mathbb{R}^{3\times2\times1}$.

As the tensor $\mathcal{X}$ is in the tensor space $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$, the tensor space can be interpreted as the Kronecker product of N vector spaces $\mathbb{R}^{I_1}, \mathbb{R}^{I_2}, \ldots, \mathbb{R}^{I_N}$. To project $\mathcal{X} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ to another tensor $\mathcal{Y}$ in a lower-dimensional tensor space $\mathbb{R}^{P_1} \otimes \mathbb{R}^{P_2} \otimes \cdots \otimes \mathbb{R}^{P_N}$, where $P_n \leq I_n$ for $n = 1, 2, \ldots, N$, we need N projection matrices $\{\boldsymbol{U}^{(n)} \in \mathbb{R}^{I_n \times P_n}\}_{n=1}^{N}$. Using the N projection matrices, the TTP is given by

$$\mathcal{Y} = \mathcal{X} \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \cdots \times_N \boldsymbol{U}^{(N)\top}. \tag{2.40}$$

This projection is established in N steps, where at the nth step, each n-mode vector is projected to a P_n-dimensional space by $\boldsymbol{U}^{(n)}$. We call this operation the orthogonal projection of $\mathcal{X}$ to $\mathcal{Y}$.

We define the inner product of two tensors $\mathcal{X} = (x_{i_1,i_2,\ldots,i_N}), \mathcal{Y} = (y_{i_1,i_2,\ldots,i_N}) \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ by

$$\langle \mathcal{X}, \mathcal{Y} \rangle \quad = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} x_{i_1,i_2,\ldots,i_N} y_{i_1,i_2,\ldots,i_N}. \tag{2.41}$$

Using this inner product, we have the Frobenius norm of a tensor $\mathcal{X}$ by

$$\|\mathcal{X}\|_{\mathrm{F}} = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}. \tag{2.42}$$

For the Frobenius norm of a tensor, we have

$$\|\mathcal{X}\|_{\mathrm{F}} = \|\mathrm{vec}\ \mathcal{X}\|_2, \tag{2.43}$$

where $\|\cdot\|_2$is Euclidean norm for a vector, respectively. For the two tensors $\mathcal{X}_1$ and $\mathcal{X}_2$, we define the distance between them by

$$d(\mathcal{X}_1, \mathcal{X}_2) = \|\mathcal{X}_1 - \mathcal{X}_2\|_{\mathrm{F}}. \tag{2.44}$$

Although this definition is a tensor-based measure, this distance is equivalent to the Euclidean distance between the vectorised tensors $\mathcal{X}_1$ and $\mathcal{X}_2$.

2.2 Principal Component Analysis

2.2.1 First-Order Tensor

Karhunen-Louéve Expansion and Transform

Setting H to be the space of data, we assume that the inner norm (f, g) between $f, g \in H$ is defined in H. For $f \in H$, we have L_2-norm $\|f\|_2 =$

$\sqrt{(f,f)}$. Furthermore, we define the Schatten product $\langle f, g\rangle$, which is an operator from H to H. We construct an operator P for $f \in H$ such that

$$P = \arg\min\left(E(\|f - Pf\|_2)\right) \quad \text{w.r.t} \quad P^*P = I, \tag{2.45}$$

where I and $E(\cdot)$ are the identity operator and the expectation over H. Setting $\{\varphi_j\}_{j=1}^{\infty}$ the eigenfunction of $M = E(\langle f, f\rangle)$, we define the principal eigenfunction of M as $\|\varphi_j\|_2 = 1$ for corresponding eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_j \geq \cdots \geq \lambda_\infty$. Using $k \in \mathbb{N}$ eigenfunctions, we define the operation $P = \sum_{j=1}^{k} \langle \varphi_j, \varphi_i\rangle$. The procedure of finding $\{\varphi_j\}_{j=1}^{\infty}$ and the projection Pf are called Karhunen-Louéve expansion and transform, respectively [84, 108].

Vector Principal Component Analysis

For practical computation of Karhunen-Louéve expansion and transform, we adopt sampled data $\{\boldsymbol{x}_i\}_{i=1}^{N}$ such that $\boldsymbol{x} \in \mathbb{R}^m$ with zero expectation $E(\boldsymbol{x}) = \boldsymbol{0}$. Furthermore, using inner product $(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top \boldsymbol{y}$ for two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, we define norm $\|\boldsymbol{x}\|_2 = \boldsymbol{x}^\top \boldsymbol{x}$ of a vector $\boldsymbol{x}$. For the sampled data, we define transform

$$\hat{\boldsymbol{x}}_i = \boldsymbol{U}^\top \boldsymbol{x}_i \tag{2.46}$$

by the orthogonal projection matrix $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_k] \in \mathbb{R}^{m\times k}$. This projection is the discretised KL transform. This projection minimises the criterion

$$J_- = E\left(\|\boldsymbol{x}_i - \boldsymbol{U}\hat{\boldsymbol{x}}\|_2^2\right). \tag{2.47}$$

This minimisation is equivalent to the maximisation of the criterion

$$\begin{aligned} J_+ &= E\left(\|\boldsymbol{U}^\top \boldsymbol{x}_i\|_2^2\right) = \sum_{i=1}^{N} (\boldsymbol{U}^\top \boldsymbol{x}_i)(\boldsymbol{U}^\top \boldsymbol{x}_i)^\top, && (2.48)\\ &= \boldsymbol{U}^\top \left(\sum_{i=1}^{N} \boldsymbol{x}\boldsymbol{x}^\top\right) \boldsymbol{U}, && (2.49)\\ &= \boldsymbol{U}^\top \boldsymbol{N}\boldsymbol{U}. && (2.50) \end{aligned}$$

Using is a identity matrix $\boldsymbol{I}$ and a set of Lagrange multipliers $\lambda_1, \lambda_2, \ldots, \lambda_d$ with condition $\lambda_1 \geq \lambda_2 \geq \ldots, \geq \lambda_m$, we have a function

$$\begin{aligned} E(\boldsymbol{U}) &= J_- - \operatorname{tr}\left((\boldsymbol{U}^\top \boldsymbol{U} - \boldsymbol{I})\boldsymbol{\Lambda}\right), && (2.51)\\ &= \operatorname{tr}\left(\boldsymbol{U}^\top \boldsymbol{N}\boldsymbol{U}\right) - \operatorname{tr}\left((\boldsymbol{U}^\top \boldsymbol{U} - \boldsymbol{I})\boldsymbol{\Lambda}\right), && (2.52) \end{aligned}$$

where we set $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)$. By maximising this function, we obtain an orthogonal matrix $\boldsymbol{U}$. To maximise the function, we compute

stationary point using partial deviation against $\boldsymbol{U}$ as [1]

$$\mathrm{tr}\left((\boldsymbol{U}+\boldsymbol{U}^\top)\boldsymbol{N}\right)-\mathrm{tr}(2\boldsymbol{U})\boldsymbol{\Lambda}=0. \tag{2.53}$$

From eq.(2.53), we have diagonalisatoin

$$\boldsymbol{U}^\top\boldsymbol{N}\boldsymbol{U}=\boldsymbol{\Lambda}. \tag{2.54}$$

Therefore, solving an eigenvalue problem

$$\boldsymbol{N}\boldsymbol{u}_i=\lambda_i\boldsymbol{u}_i, \tag{2.55}$$

we obtain an orthogonal matrix $\boldsymbol{U}$. This is the discretised KL expansion. The results obtained by these practical computation of discretised KL expansion and transform are called principal component analysis. For the computation for the eigenvalue problem, time complexity is $\mathcal{O}(m^3)$.

Since $\lambda_j \ll \lambda_1$ for $j > k$, using only k eigenvectors, we have dimension reduction satisfying eq.(2.47). Using an orthogonal projection matrix $\boldsymbol{P} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_k]$, $\boldsymbol{e}_i \in \mathbb{R}^m$, we have the dimension reduction of $\boldsymbol{x}_i$ as

$$\hat{\boldsymbol{x}}_i=(\boldsymbol{P}\boldsymbol{U})^\top\boldsymbol{x}_i, \tag{2.56}$$

where $\boldsymbol{P}$ selects k major eigenvectors, which are correspond to k large eigenvalues in the order of decreasing, as bases for a projection. This dimension reduction is the reduction by the principal component analysis (PCA).

2.2.2 Second-Order Tensor

Two-Dimensional Singular Value Decomposition

For a collection of matrices $\{\boldsymbol{X}_i\}_{i=1}^N \in \mathbb{R}^{m\times n}$ satisfying the zero expectation condition $\mathrm{E}(\boldsymbol{X}_i) = 0$, the orthogonal-projection-based data-reduction mapping

$$\hat{\boldsymbol{X}}_i=\boldsymbol{U}^\top\boldsymbol{X}_i\boldsymbol{V}, \tag{2.57}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m]$ and $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n]$, is performed by minimising the criterion

$$J_-=\mathrm{E}\left(\|\boldsymbol{X}_i-\boldsymbol{U}\hat{\boldsymbol{X}}_i\boldsymbol{V}^\top\|_{\mathrm{F}}^2\right) \tag{2.58}$$

and maximising the criterion

$$J_+ = \mathrm{E}\left(\|\boldsymbol{U}^\top\boldsymbol{X}_i\boldsymbol{V}\|_{\mathrm{F}}^2\right), \tag{2.59}$$

$$\tag{2.60}$$

[1] Here, $\frac{\partial}{\partial\boldsymbol{U}}\boldsymbol{U}^\top\boldsymbol{M}\boldsymbol{U}=(\boldsymbol{U}+\boldsymbol{U}^\top)$ and $\frac{\partial}{\partial\boldsymbol{U}}\boldsymbol{U}^\top\boldsymbol{U}=2\boldsymbol{U}$.

with respect to the conditions

$$\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}_m,\ \boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{I}_n, \tag{2.61}$$

where $\boldsymbol{I}_m$ and $\boldsymbol{I}_n$ are the identity matrices in $\mathbb{R}^{m\times m}$ and $\mathbb{R}^{n\times n}$, respectively. The minimisation of eq. (2.58) is equivalent to the maximisation of eq. (2.59). Furthermore, by fixing one of $\boldsymbol{V}$ and $\boldsymbol{U}$ in this maximisation, we derive the maximisation criterion

$$\begin{aligned} J_V &= \mathrm{E}\left(\|\boldsymbol{V}^\top \boldsymbol{X}_i^\top \boldsymbol{X}_i \boldsymbol{V}\|_{\mathrm{F}}^2\right), & (2.62)\\ J_U &= \mathrm{E}\left(\|\boldsymbol{U}^\top \boldsymbol{X}_i \boldsymbol{X}_i^\top \boldsymbol{U}\|_{\mathrm{F}}^2\right) & (2.63) \end{aligned}$$

with respect to the conditions of eq. (2.61).

From J_-, J_+, J_V and J_U, the eigendecomposition problems are derived by computing the extremals of

$$\begin{aligned} E_- &= J_- + tr((\boldsymbol{I} - \boldsymbol{V}^\top \boldsymbol{V})\boldsymbol{\Lambda}) + tr((\boldsymbol{I} - \boldsymbol{U}^\top \boldsymbol{U})\boldsymbol{\Sigma}), & (2.64)\\ E_+ &= J_+ + tr((\boldsymbol{I} - \boldsymbol{V}^\top \boldsymbol{V})\boldsymbol{\Lambda}) + tr((\boldsymbol{I} - \boldsymbol{U}^\top \boldsymbol{U})\boldsymbol{\Sigma}), & (2.65)\\ E_V &= J_V + tr((\boldsymbol{I} - \boldsymbol{V}^\top \boldsymbol{V})\boldsymbol{\Lambda}), & (2.66)\\ E_U &= J_U + tr((\boldsymbol{I} - \boldsymbol{U}^\top \boldsymbol{U})\boldsymbol{\Sigma}), & (2.67) \end{aligned}$$

respectively. To optimise both J_- and J_+, we set $\boldsymbol{M} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{X}_i^\top \boldsymbol{X}_i$ and $\boldsymbol{N} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{X}_i \boldsymbol{X}_i^\top$, where $K = \mathrm{rank}(\boldsymbol{M}) = \mathrm{rank}(\boldsymbol{N})$. Then, from the optimisations of J_V and J_U, we derive the eigenvalue problems

$$\begin{aligned} \boldsymbol{M}\boldsymbol{V} &= \boldsymbol{V}\boldsymbol{\Lambda}, & (2.68)\\ \boldsymbol{N}\boldsymbol{U} &= \boldsymbol{U}\boldsymbol{\Sigma}, & (2.69) \end{aligned}$$

where $\boldsymbol{\Sigma} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{\Lambda} \in \mathbb{R}^{n\times n}$ are diagonal matrices satisfying the relationships $\lambda_i = \sigma_i$ for

$$\begin{aligned} \boldsymbol{\Sigma} &= \mathrm{diag}(\sigma_1, \sigma_2 \cdots, \sigma_K, 0 \cdots, 0), & (2.70)\\ \boldsymbol{\Lambda} &= \mathrm{diag}(\lambda_1, \lambda_2 \cdots, \lambda_K, 0 \cdots, 0). & (2.71) \end{aligned}$$

The optimisations of J_V and J_U derive the SVD problems of eqs. (2.68) and (2.69), respectively [2]. For $\boldsymbol{p}_{1j} \in \{\boldsymbol{e}_j\}_{j=1}^K$ and $\boldsymbol{p}_{2j} \in \{\boldsymbol{e}_k\}_{k=1}^K, \boldsymbol{e}_i^\top \boldsymbol{e}_j = \delta_{ij}$, we set the orthogonal projection matrices $\boldsymbol{P}_1 = \sum_{j=1}^{k_1} \boldsymbol{p}_{1j}\boldsymbol{p}_{1j}^\top$ and $\boldsymbol{P}_2 = \sum_{j=1}^{k_2} \boldsymbol{p}_{2j}\boldsymbol{p}_{2j}^\top$. Using these $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, the low-rank matrix approximation [104] is given by

$$\boldsymbol{Y}_i = (\boldsymbol{U}\boldsymbol{P}_1)^\top \boldsymbol{X}_i (\boldsymbol{V}\boldsymbol{P}_2) = \boldsymbol{L}^\top \boldsymbol{X}_i \boldsymbol{R}, \tag{2.72}$$

[2] For an iterative method for two-dimensional singular value decomposition see refs. [72, 125].

where $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are the k_1 and k_2 selected basis vectors of projection matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively. The low-rank approximation using eq. (2.72) is called the two-dimensional singular value decomposition (2DSVD) method in the context of image compression [1, 44]. Furthermore, the method based on the transform

$$\boldsymbol{Y}_i = \boldsymbol{X}_i \boldsymbol{R} \tag{2.73}$$

is called the two-dimensional principal component analysis [184].

For the minimisation of eq. (2.58), and maximisation of eqs. (2.59), (2.62) and (2.63), as generalised principal component analysis (GPCA), the iterative algorithm [185] described in Algorithm 1.1 is proposed. The original iterative algorithm of the GPCA used identity matrices as the initial projection matrices in step 1. This initialisation increase the number of iterations without any effects since the computation in step 4 in the first iteration corresponds to the marginal eigenvalue (MEV). On the other hand, using the projection matrices obtained by the higher-order singular value decomposition (HOSVD) as the initial projection matrices, the iterative algorithm in the multilinear principal component analysis (MPCA) converges within three iterations for three-dimensional data [112]. For the 2DSVD, using the projection matrices obtained by the MEV as the initial projection of the GPCA, we can reduce the number of iterations. Therefore, Algorithm 1.1 is a refined version of the GPCA with the MEV in step 1. In Algorithm 1.1, if the dimensions of a projected matrix are coincident to these of an original matrix, the obtained projection is called a full projection, otherwise, the projection called a full projection truncation.

For Algorithm 1.1, we have the following property.

Property 2.1 *The full projection of the GPCA without iterations is equivalent to the projection by the MEV.*

For the 2DSVD, we have the following theorem

Theorem 2.2 *The 2DSVD method is equivalent to the classical PCA method. (Proof) The equation*

$$(\boldsymbol{U}\boldsymbol{P}_1)^\top \boldsymbol{X}(\boldsymbol{V}\boldsymbol{P}_2) = \boldsymbol{Y} \tag{2.74}$$

is equivalent to

$$(\boldsymbol{V}\boldsymbol{P}_2 \otimes \boldsymbol{U}\boldsymbol{P}_1)\mathrm{vec}\boldsymbol{X} = \mathrm{vec}\boldsymbol{Y}. \tag{2.75}$$

(Q.E.D.)

Algorithm 1.1: Iterative method in the GPCA

Input: A set of tensors $\{\boldsymbol{X}_i \in \mathbb{R}^{m\times n}\}_{i=1}^{N}$. Reduced dimensions k_1 and k_2 for modes 1 and 2, respectively. The maximum number of iterations K.

Output: A set of projection matrices $\{\boldsymbol{P}_{\mathrm{L}}, \boldsymbol{P}_{\mathrm{R}}\}$. If $k_1 = m$ and $k_2 = n$, $\{\boldsymbol{P}_{\mathrm{L}}, \boldsymbol{P}_{\mathrm{R}}\}$ gives full projection, otherwise, it gives full projection truncation.

1: Compute the initial projection matrices $\boldsymbol{P}_{\mathrm{L}}^{(0)}$ and $\boldsymbol{P}_{\mathrm{R}}^{(0)}$ by the eigendecompositions of $\boldsymbol{M}_r^{(0)} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{X}_i\boldsymbol{X}_i^{\top}$ and $\boldsymbol{M}_c^{(0)} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{X}_i^{\top}\boldsymbol{X}_i$, respectively.

2: Construct projection matrices for each mode by selecting k_1 and k_2 eigenvectors of $\boldsymbol{M}_r^{(0)}$ and $\boldsymbol{M}_c^{(0)}$, respectively, corresponding to the k_j largest eigenvalues for $j = 1, 2$.

3: Compute $\Psi_0 = \sum_{i=1}^{N} \|\boldsymbol{P}_{\mathrm{L}}^{(0)\top}\boldsymbol{X}_i\boldsymbol{P}_{\mathrm{R}}^{(0)}\|_{\mathrm{F}}^2$.

4: Begin loop

for $k = 1, 2, \ldots, K$

Compute $\boldsymbol{P}_{\mathrm{L}}^{(k)}$ by selecting k_1 eigenvectors from the eigendecomposition for a matrix $\boldsymbol{M}_r^{(k)} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{X}_i\boldsymbol{P}_{\mathrm{R}}^{(k-1)}\boldsymbol{P}_{\mathrm{R}}^{(k-1)\top}\boldsymbol{X}_i^{\top}$.

Compute $\boldsymbol{P}_{\mathrm{R}}^{(k)}$ by selecting k_2 eigenvectors of the eigendecomposition for a matrix $\boldsymbol{M}_c^{(k)} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{X}_i^{\top}\boldsymbol{P}_{\mathrm{L}}^{(k-1)\top}\boldsymbol{P}_{\mathrm{L}}^{(k-1)}\boldsymbol{X}_i$.

if $|\Psi_k - \Psi_{k-1}| < \eta$, where $\Psi_k = \sum_{j=1}^{M} \|\boldsymbol{P}_{\mathrm{L}}^{(k)\top}\boldsymbol{X}_i\boldsymbol{P}_{\mathrm{R}}^{(k)}\|_{\mathrm{F}}^2$, break

end

5: Return $\boldsymbol{P}_{\mathrm{L}} = \boldsymbol{P}_{\mathrm{L}}^{(k)}$ and $\boldsymbol{P}_{\mathrm{R}} = \boldsymbol{P}_{\mathrm{R}}^{(k)}$

The eigenfunction and eigendistribution of the two-dimensional discrete cosine transform (2DDCT) [3] approximately coincide with those of the Karhunen-Loeve expansion for images. In special cases, the reduction using the 2DDCT is identical to the reduction using the PCA. Figure 2.8 illustrates the representation of an image and the reduction by the 2DDCT, the PCA and the second-order tensor principal component analysis (TPCA). The 2DDCT and the PCA are unitary transforms; therefore, their bases are related to a rotation transformation. Furthermore, the 2DDCT is an acceptable approximation of the 2DSVD, since the reduction using the 2DDCT is an acceptable approximation of the reduction using the PCA [79]. In this case, For the computation of Algorithm 1.1, time complexity is $\mathcal{O}(Km^3)$, where $m \geq n$, and K is iteration number. Moreover, the projection that selects the $K = k_1 k_2$ basis of the tensor space spanned by $\boldsymbol{u}_i \otimes \boldsymbol{v}_j$, $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, is

$$(\boldsymbol{V}\boldsymbol{P}_2 \otimes \boldsymbol{U}\boldsymbol{P}_1) = (\boldsymbol{V} \otimes \boldsymbol{U})(\boldsymbol{P}_2 \otimes \boldsymbol{P}_1) = \boldsymbol{W}\boldsymbol{P}, \tag{2.76}$$

where $\boldsymbol{W}$ and $\boldsymbol{P}$ are a orthogonal matrix and the orthogonal projection matrix, respectively. Therefore, the 2DSVD is equivalent to the TPCA for matrices because matrices are second-order tensors.

Marginal Eigenvectors

We define two matrices

$$\boldsymbol{M}_{\mathrm{r}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{X}_i \boldsymbol{X}_i^{\top} \in \mathbb{R}^{m \times m}, \tag{2.77}$$

$$\boldsymbol{M}_{\mathrm{c}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{X}_i^{\top} \boldsymbol{X}_i \in \mathbb{R}^{n \times n}. \tag{2.78}$$

Using these two matrices, we have

$$\begin{aligned} \mathrm{E}\left(\|\boldsymbol{P}_{\mathrm{L}}^{\top}\boldsymbol{X}_i\boldsymbol{P}_{\mathrm{R}}\|_{\mathrm{F}}^2\right) &= \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{P}_{\mathrm{L}}^{\top}\boldsymbol{X}_i\boldsymbol{P}_{\mathrm{R}}\right)\left(\boldsymbol{P}_{\mathrm{L}}^{\top}\boldsymbol{X}_i\boldsymbol{P}_{\mathrm{R}}\right)^{\top} \\ &= \boldsymbol{P}_{\mathrm{L}}^{\top}\left(\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{X}_i\boldsymbol{X}_i^{\top}\right)\boldsymbol{P}_{\mathrm{L}}, \\ &= \boldsymbol{P}_{\mathrm{L}}^{\top}\boldsymbol{M}_{\mathrm{r}}\boldsymbol{P}_{\mathrm{L}}, \end{aligned} \tag{2.79}$$

[3] Note that we use the two-dimensional DCT-II without dividing an image into blocks, while the JPEG and MPEG compression algorithms use the two-dimensional DCT-II by partitioning an image into $N \times N$ blocks of 8×8 pixels.

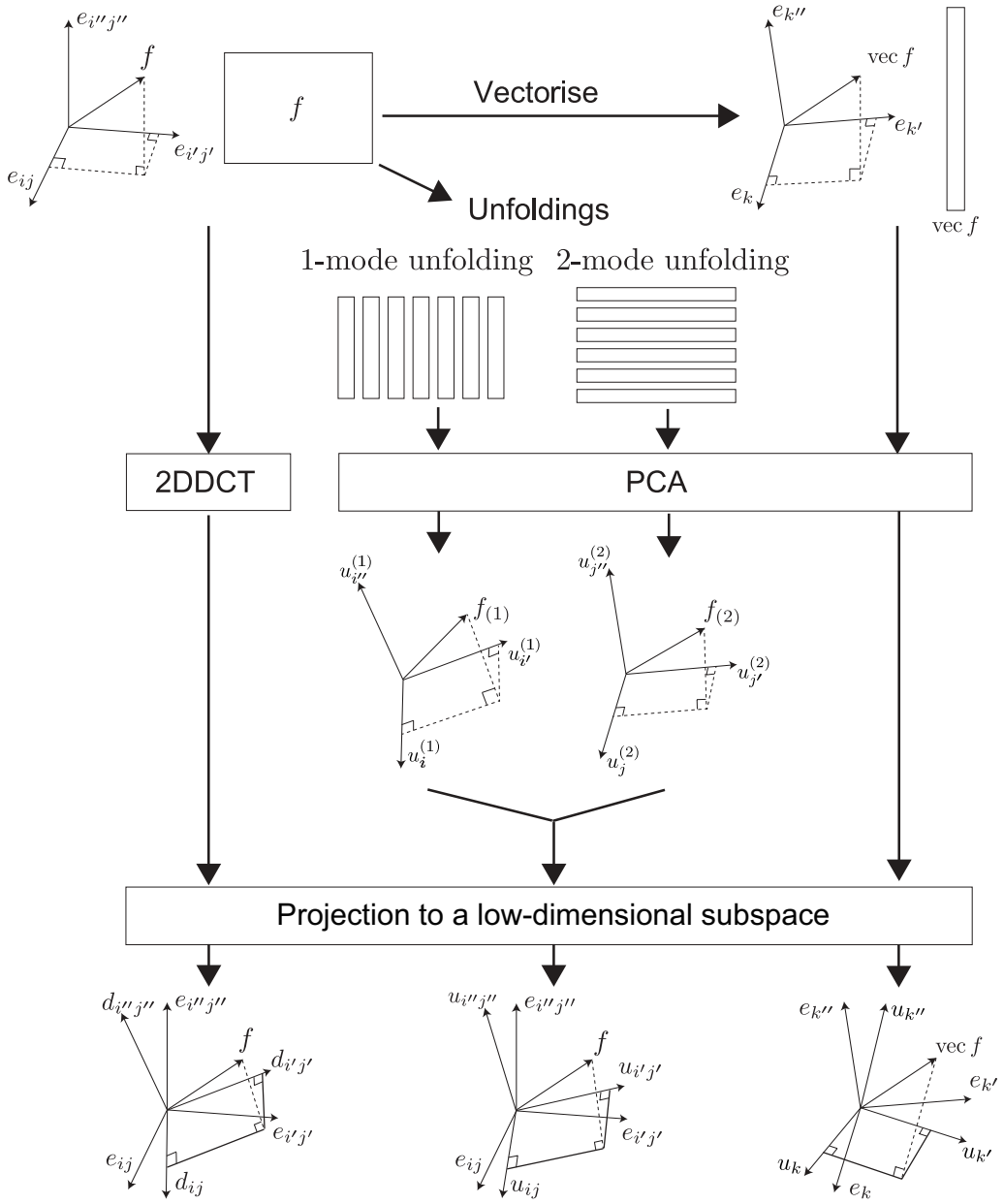


Figure 2.8: Image representation and dimension reduction. In the origin of the flow, $e_{ij}, e_{i'j'}, e_{i''j''}$ are the basis representing each pixel of an image f. After vectorisation of the image, $e_k, e_{k'}, e_{k''}$ are the standard basis for the Euclidean space for the vectorised image vec f. After 1-mode unfolding, $u^{(1)}_i, u^{(1)}_{i'}$ and $u^{(1)}_{i''}$ are the basis of the TPCA for the 1-mode unfolded image $f_{(1)}$. After 2-mode unfoldings, $u^{(2)}_j, u^{(2)}_{j'}$ and $u^{(2)}_{j''}$ are the basis of the TPCA for the 2-mode unfolded image $f_{(2)}$. $d_{ij}, d_{i'j'}, d_{i''j''}$ are the basis of the 2DDCT. After the PCA for the vectorised image, $u_k, u_{k'}, u_{k''}$ are the basis of the PCA. After the PCA for the 1- and 2-mode unfolded image, $u_{ij}, u_{i'j'}, u_{i''j''}$ are the basis of the 2D tensor space. Here, $u_{ij} = u^{(1)}_i \otimes u^{(2)}_j$, $u_{i'j'} = u^{(1)}_{i'} \otimes u^{(2)}_{j'}$ and $u_{ij} = u^{(1)}_{i''} \otimes u^{(2)}_{j''}$. By selecting the basis, we obtain an orthogonal projection to a lower-dimensional subspace.

and

$$\begin{aligned}\mathrm{E}\left(\|\boldsymbol{P}_{\mathrm{L}}^{\top}\boldsymbol{X}_i\boldsymbol{P}_{\mathrm{R}}\|_{\mathrm{F}}^2\right) &= \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{P}_{\mathrm{L}}^{\top}\boldsymbol{X}_i\boldsymbol{P}_{\mathrm{R}}\right)^{\top}\left(\boldsymbol{P}_{\mathrm{L}}^{\top}\boldsymbol{X}_i\boldsymbol{P}_{\mathrm{R}}\right)\\ &= \boldsymbol{P}_{\mathrm{R}}^{\top}\left(\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{X}_i^{\top}\boldsymbol{X}_i\right)\boldsymbol{P}_{\mathrm{R}},\\ &= \boldsymbol{P}_{\mathrm{R}}^{\top}\boldsymbol{M}_{\mathrm{C}}\boldsymbol{P}_{\mathrm{R}}. \end{aligned} \tag{2.80}$$

Furthermore, for the two matrices $\boldsymbol{M}_{\mathrm{r}}$ and $\boldsymbol{M}_{\mathrm{c}}$, using the Lagrange multipliers Λ_{r} and Λ_{c}, we find projections satisfying

$$J(\boldsymbol{P}_{\mathrm{L}}) = tr\left(\boldsymbol{P}_{\mathrm{L}}^{\top}\boldsymbol{M}_{\mathrm{r}}\boldsymbol{P}_{\mathrm{L}}\right) - tr\left(\left(\boldsymbol{P}_{\mathrm{L}}^{\top}\boldsymbol{P}_{\mathrm{L}} - \boldsymbol{I}\right)\Lambda_{\mathrm{r}}\right), \tag{2.81}$$

$$J(\boldsymbol{P}_{Y}) = tr\left(\boldsymbol{P}_{\mathrm{R}}^{\top}\boldsymbol{M}_{\mathrm{c}}\boldsymbol{P}_{\mathrm{R}}\right) - tr\left(\left(\boldsymbol{P}_{\mathrm{R}}^{\top}\boldsymbol{P}_{\mathrm{R}} - \boldsymbol{I}\right)\Lambda_{\mathrm{c}}\right), \tag{2.82}$$

where $\boldsymbol{I}$ is the identity matrix. The solutions of eqs. (2.81) and (2.82) are given as the solutions of the eigenproblems of $\boldsymbol{M}_{\mathrm{r}}$ and $\boldsymbol{M}_{\mathrm{c}}$, respectively. We set $\{\boldsymbol{u}_j\}_{j=1}^{k_1}$ and $\{\boldsymbol{v}_j\}_{j=1}^{k_2}$ as the eigenvectors of $\boldsymbol{M}_{\mathrm{r}}$ and $\boldsymbol{M}_{\mathrm{c}}$, respectively. We define the eigenvectors of $\boldsymbol{M}_{\mathrm{r}}$ and $\boldsymbol{M}_{\mathrm{c}}$ as $\|\boldsymbol{u}_j\|_2 = 1$ and $\|\boldsymbol{v}_j\|_2 = 1$ for eigenvalues $\lambda_1^{\mathrm{r}} \geq \lambda_2^{\mathrm{r}} \geq \cdots \geq \lambda_j^{\mathrm{r}} \geq \cdots \geq \lambda_n^{\mathrm{r}}$ and $\lambda_1^{\mathrm{c}} \geq \lambda_2^{\mathrm{c}} \geq \cdots \geq \lambda_j^{\mathrm{c}} \geq \cdots \geq \lambda_n^{\mathrm{c}}$, respectively. Therefore, for given numbers $k_1 \leq m$ and $k_2 \leq n$, the operators $\boldsymbol{P}_{\mathrm{L}}$ and $\boldsymbol{P}_{\mathrm{R}}$ are defined as $\boldsymbol{P}_{\mathrm{L},k_1} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_m]$ and $\boldsymbol{P}_{\mathrm{R},k_2} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n]$ as the matrices consist of each set of eigenfunctions, respectively. These obtained projections are equivalent to the projections in the eq. (2.72) obtained by 2DSVD [1] shown in the previous subsection. The time complexity of the MEV is $mathcalO(m^3)$ for $m \geq n$.

2.2.3 Third-Order Tensor

A third-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, which is the array $\boldsymbol{X} = (x_{i_1,i_2,i_3}) \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, is denoted as a triple of indices (i_1, i_2, i_3). Here we summarise the HOSVD for third-order tensors. For a collection of tensors $\{\mathcal{X}_i\}_{i=1}^{N} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ satisfying the zero expectation condition $\mathrm{E}(\mathcal{X}_i) = 0$, we compute the

$$\hat{\mathcal{X}}_i = \mathcal{X}_i \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \times_3 \boldsymbol{U}^{(3)\top}, \tag{2.83}$$

where $\boldsymbol{U}^{(j)} = [\boldsymbol{u}_1^{(j)}, \ldots, \boldsymbol{u}_{I_j}^{(j)}]$, that minimises the criterion

$$J_{-} = \mathrm{E}\left(\|\mathcal{X}_i - \hat{\mathcal{X}}_i \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \boldsymbol{U}^{(3)}\|_{\mathrm{F}}^2\right) \tag{2.84}$$

and maximises the criteria

$$J_{+} = \mathrm{E}\left(\|\hat{\mathcal{X}}_i\|_{\mathrm{F}}^2\right), \tag{2.85}$$

with respect to the conditions

$$\boldsymbol{U}^{(j)\top}\boldsymbol{U}^{(j)} = \boldsymbol{I}_j, \tag{2.86}$$

where $\boldsymbol{I}_j,\ j = 1, 2, 3$ is the identity matrices in $\mathbb{R}^{I_j \times I_j}$. For these criterion, by fixing two of $\boldsymbol{U}^{(1)}$, $\boldsymbol{U}^{(2)}$ and $\boldsymbol{U}^{(3)}$, we have the following criteria

$$J_j = \mathrm{E}\left(\|\boldsymbol{U}^{(j)\top}\mathcal{X}_{i,(j)}\mathcal{X}_{i,(j)}^{\top}\boldsymbol{U}^{(j)}\|_{\mathrm{F}}^2\right), \tag{2.87}$$

where $\mathcal{X}_{i,(j)},\ j = 1, 2, 3$, are the j-mode unfolded tensor $\mathcal{X}_i$, with respect to eq. (2.86).

Eigendecomposition problems are derived by computing the extremal of

$$E_- \ = \ J_- + \sum_{j=1}^{N} tr((\boldsymbol{I}_j - \boldsymbol{U}^{(j)\top}\boldsymbol{U}^{(j)})\boldsymbol{\Sigma}^{(j)}). \tag{2.88}$$

As an extension of the two-dimensional problem, we define the system of minimisation problems

$$E_j \ = \ J_j + tr((\boldsymbol{I}_j - \boldsymbol{U}^{(j)\top}\boldsymbol{U}^{(j)})\boldsymbol{\Sigma}^{(j)}),\ j = 1, 2, 3. \tag{2.89}$$

For matrices $\boldsymbol{M}^{(j)} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{X}_{i,(j)}\mathcal{X}_{i,(j)}^{\top},\ j = 1, 2, 3$, the optimisation of J_- derives the eigenvalue decomposition

$$\boldsymbol{M}^{(j)}\boldsymbol{U}^{(j)} = \boldsymbol{U}^{(j)}\boldsymbol{\Sigma}^{(j)}, \tag{2.90}$$

where $\boldsymbol{\Sigma}^{(j)} \in \mathbb{R}^{I_j \times I_j},\ j = 1, 2, 3$, are diagonal matrices satisfying the relationships $\sigma_k^{(j)} = \sigma_k^{(j')},\ k \in \{1, 2, \ldots, K\},\ K = \mathrm{rank}(\boldsymbol{M}^{(1)}) = \mathrm{rank}(\boldsymbol{M}^{(2)}) = \mathrm{rank}(\boldsymbol{M}^{(3)})$ for

$$\boldsymbol{\Sigma}^{(j)} \ = \ \mathrm{diag}(\lambda_1^{(j)}, \lambda_2^{(j)} \cdots, \lambda_K^{(j)}, 0 \cdots, 0). \tag{2.91}$$

The optimisation of each J_j derives the eigendecomposition problems in eq. (2.90). However, for the optimisation of $\{J_j\}_{j=1}^3$, there is no closed-form solution to this maximisation problem [100, 99]. Algorithm 1.2 is the iterative procedure of the MPCA. For Algorithm 1.2, we have the following property.

Proposition 2.1 *The MPCA without iteration is equivalent to the HOSVD if dimensions of a projected tensor are coincident to ones of each mode of an original tensor.*

For third-order tensors, there are 3! combinations in selecting the order of modes ξ_1, ξ_2 and ξ_3 in a tensor-to-tensor projection for Algorithm 1.2. On the other hand, one combination exists in selecting the order of modes for the second-order tensors.

Algorithm 1.2:Iterative method in the MPCA for third-order tensors

Input: A set of tensors $\{\mathcal{X}_i\}_{i=1}^N$. Dimension of projected tensors $\{k_j\}_{j=1}^3$. A maximum number of iteration K. An order ξ_1, ξ_2, ξ_3 to select the unfolded tensors. A sufficiently small number η.

Output: A set of projection matrices $\{\boldsymbol{U}^{(j)}\}_{j=1}^3$.

1: Compute the eigendecomposition of a covariant matrix $\boldsymbol{M}^{(j)} = \frac{1}{N}\sum_{i=1}^N \mathcal{X}_{i,(j)}\mathcal{X}_{i,(j)}^\top$, where $\mathcal{X}_{i,(j)}$ is an j-mode unfolded $\mathcal{X}_i$, for $j = 1, 2, 3$.

2: Construct projection matrices by selecting eigenvectors corresponding to the k_j largest eigenvalues for $j = 1, 2, 3$.

3: Compute $\Psi_0 = \sum_{i=1}^N \|\mathcal{X}_i \times_{\xi_1} \boldsymbol{U}^{(\xi_1)\top} \times_{\xi_2} \boldsymbol{U}^{(\xi_2)\top} \times_{\xi_3} \boldsymbol{U}^{(\xi_3)\top}\|_{\mathrm{F}}$.

4: Iteratively Compute the following procedure.

for $k = 1, 2, \ldots, K$

for $j = \xi_1, \xi_2, \xi_3$

Update $\boldsymbol{U}^{(j)}$ by decomposing matrix $\boldsymbol{M}^{(j)} = \sum_{i=1}^N \mathcal{W}_i, (j)\mathcal{W}_{i,(j)}^\top$, where $\mathcal{W}_{i,(j)}$ is an j-mode unfolded $\mathcal{W}_i = \mathcal{X}_i \times_{\xi_\alpha} \boldsymbol{U}^{(\xi_\alpha)\top} \times_{\xi_\beta} \boldsymbol{U}^{(\xi_\beta)\top}$ for $\xi_\alpha, \xi_\beta \in \{\xi_1, \xi_2, \xi_3\} \setminus j$, $\xi_\alpha \neq \xi_\beta$.

end

Compute $\Psi_k = \sum_{i=1}^N \|\mathcal{X}_i \times_{\xi_1} \boldsymbol{U}^{(\xi_1)\top} \times_{\xi_2} \boldsymbol{U}^{(\xi_2)\top} \times_{\xi_3} \boldsymbol{U}^{(\xi_3)\top}\|_{\mathrm{F}}$

if $|\Psi_k - \Psi_{k-1}| < \eta$

break

end

Proposition 2.2 *For third-order tensors, the selection of order of modes does not effect to the results of a tensor-to-tensor projection.*

From these two properties, we adopt Algorithm 1.2 [112] to solve the optimisation of $\{J_j\}_{j=1}^3$. For a set of orthonormal vectors $\{\boldsymbol{e}_k\}_{k=1}^K$, where only kth element of $\boldsymbol{e}_k$ is 1 and others are 0, we set orthogonal projection matrices $\boldsymbol{P}^{(j)} = \sum_{k=1}^{k_j} \boldsymbol{e}_k\boldsymbol{e}_k^\top$ for $j = 1, 2, 3$. Using these $\{\boldsymbol{P}^{(j)}\}_{j=1}^3$, the low-rank tensor approximation [99] is achieved by

$$\mathcal{Y} = \mathcal{X} \times_1 (\boldsymbol{P}^{(1)}\boldsymbol{U}^{(1)})^\top \times_2 (\boldsymbol{P}^{(2)}\boldsymbol{U}^{(2)})^\top \times_3 (\boldsymbol{P}^{(3)}\boldsymbol{U}^{(3)})^\top, \tag{2.92}$$

where $\boldsymbol{P}^{(j)}$ selects k_j bases of orthogonal matrices $\boldsymbol{U}^{(j)}$. The low-rank approximation using eq. (2.92) is used for compression in the TPCA.

For the HOSVD for third-order tensors, we have the following theorem.

Theorem 2.3 *The HOSVD method is equivalent to the vector PCA method.*

(*Proof*) The equation

$$\mathcal{X} \times_1 (\boldsymbol{P}^{(1)}\boldsymbol{U}^{(1)})^\top \times_2 (\boldsymbol{P}^{(2)}\boldsymbol{U}^{(2)})^\top \times_3 (\boldsymbol{P}^{(3)}\boldsymbol{U}^{(3)})^\top = \mathcal{Y} \tag{2.93}$$

is equivalent to

$$(\boldsymbol{P}^{(3)}\boldsymbol{U}^{(3)} \otimes \boldsymbol{P}^{(2)}\boldsymbol{U}^{(2)} \otimes \boldsymbol{P}^{(1)}\boldsymbol{U})^\top \mathrm{vec}\mathcal{X} = \mathrm{vec}\mathcal{Y}. \tag{2.94}$$

(Q.E.D.)

This theorem implies that the 3DDCT is an acceptable approximation of the HOSVD for third-order tensors since this is the analogy of the approximation of the PCA by the 2DDCT [130].

Furthermore, we have the following theorem.

Theorem 2.4 *The compression computed by the HOSVD is equivalent to the compression computed by the TPCA.*

(*Proof*) The projection that selects $K = k_1k_2k_3$ bases of the tensor space spanned by $u_{i_1}^{(1)} \circ u_{i_2}^{(2)} \circ u_{i_3}^{(3)},\ i_j = 1, 2, \ldots, k_j$ for $j = 1, 2, 3$, is

$$\begin{aligned}&(\boldsymbol{P}^{(3)}\boldsymbol{U}^{(3)} \otimes \boldsymbol{P}^{(2)}\boldsymbol{U}^{(2)} \otimes \boldsymbol{P}^{(1)}\boldsymbol{U}^{(1)})\\ &\quad = (\boldsymbol{P}^{(3)} \otimes \boldsymbol{P}^{(2)} \otimes \boldsymbol{P}^{(1)})(\boldsymbol{U}^{(3)} \otimes \boldsymbol{U}^{(2)} \otimes \boldsymbol{U}^{(1)}) = \boldsymbol{P}\boldsymbol{W},\end{aligned} \tag{2.95}$$

where $\boldsymbol{W}$ and $\boldsymbol{P}$ are an orthogonal matrix and a projection matrix, respectively. Therefore, HOSVD is equivalent to TPCA for third-order tensors. Figure 2.9 illustrates volumetric image representation in these different bases. If we use the MPCA and HOSVD, time complexities are $\mathcal{K}\Updownarrow^{\ni}$ and $\Updownarrow^{\ni}$, respectively, where $m \geq n, l$ for tensors of the size $m \times n \times l$.

2.2.4 Nth-Order Tensor

A Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, which is the array $\boldsymbol{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, is denoted as a set of indices $(i_1, i_2, \ldots, i_N)$. Here we summarise the higher-order singular value decomposition (HOSVD) for Nth-order tensors since N-way principal component is numerically computed by HOSVD. The HOSVD is the Tucker-3 decomposition [160] with orthogonal constraints. For a collection of tensors $\{\mathcal{X}_i\}_{i=1}^M \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ satisfying the zero expectation condition $\mathrm{E}(\mathcal{X}_i) = 0$, we compute the

$$\hat{\mathcal{X}}_i = \mathcal{X}_i \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \cdots \times_N \boldsymbol{U}^{(N)\top}, \tag{2.96}$$

where $\boldsymbol{U}^{(j)} = [\boldsymbol{u}_1^{(j)}, \ldots, \boldsymbol{u}_{I_j}^{(j)}]$, that minimises the criterion

$$J_- = \mathrm{E}\left(\|\mathcal{X}_i - \hat{\mathcal{X}}_i \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \cdots \times_N \boldsymbol{U}^{(N)}\|_{\mathrm{F}}^2\right) \tag{2.97}$$

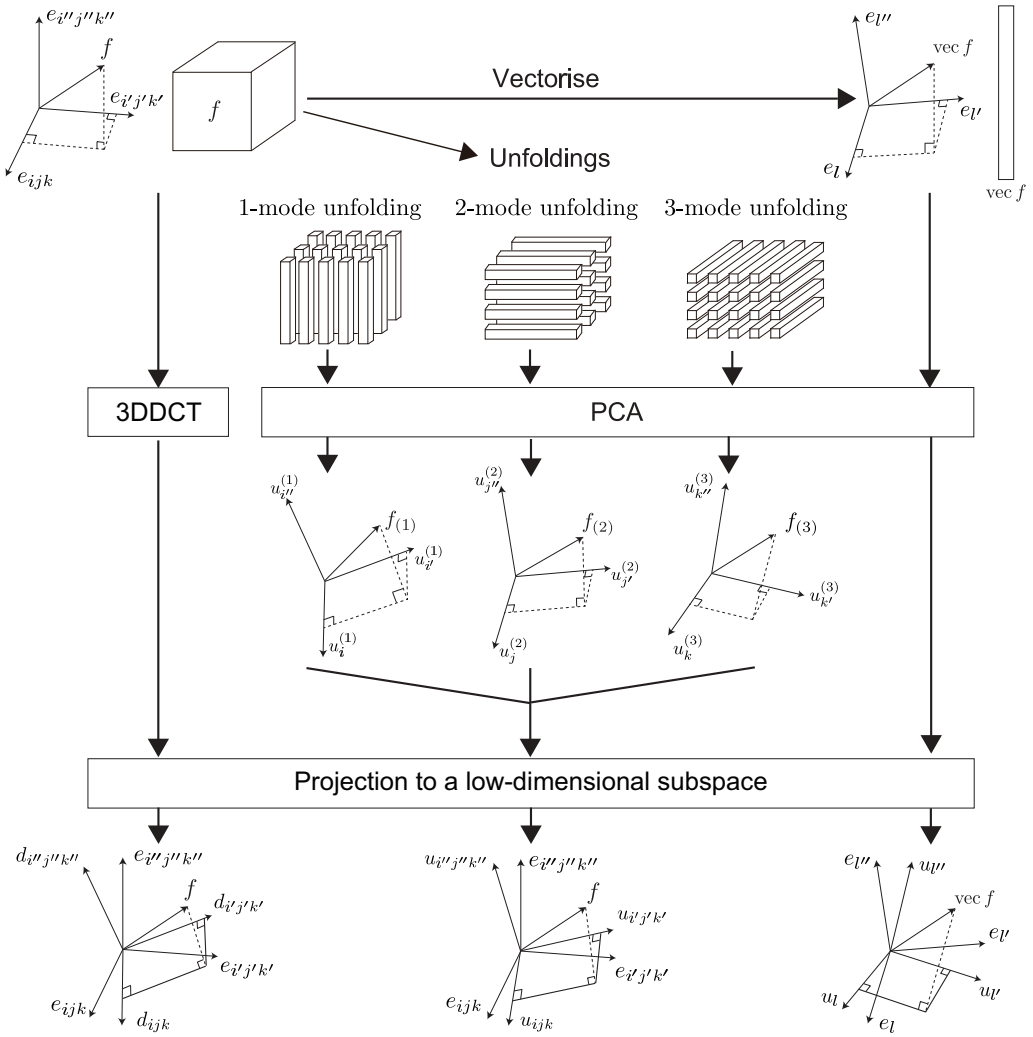


Figure 2.9: Volumetric image representation and dimension reduction. In the origin of the flow, $e_{ijk}, e_{i'j'k'}, e_{i''j''k''}$ are the basis representing each pixel of an image f. After vectorisation of the image, $e_l, e_{l'}, e_{l''}$ are the standard basis for the Euclidean space for the vectorised image $\mathrm{vec}\, f$. After 1-mode unfolding, $u_i^{(1)}, u_{i'}^{(1)}$ and $u_{i''}^{(1)}$ are the basis of the TPCA for the 1-mode unfolded image $f_{(1)}$. After 2-mode unfoldings, $u_j^{(2)}, u_{j'}^{(2)}$ and $u_{j''}^{(2)}$ are the basis of the TPCA for the 2-mode unfolded image $f_{(2)}$. After 3-mode unfoldings, $u_k^{(3)}, u_{k'}^{(3)}$ and $u_{k''}^{(3)}$ are the basis of the TPCA for the 3-mode unfolded image $f_{(3)}$. $d_{ijk}, d_{i'j'k'}, d_{i''j''k''}$ are the basis of the 3DDCT. After the PCA for the vectorised image, $u_l, u_{l'}, u_{l''}$ are the basis of the PCA. After the PCA for the 1-, 2- and 3-mode unfolded images, $u_{ijk}, u_{i'j'k'}, u_{i''j''k''}$ are the basis of the 3D tensor space. Here, $u_{ijk} = u_i^{(1)} \otimes u_j^{(2)} \otimes u_k^{(3)}$, $u_{i'j'k'} = u_{i'}^{(1)} \otimes u_{j'}^{(2)} \otimes u_{k'}^{(3)}$, $u_{i''j''k''} = u_{i''}^{(1)} \otimes u_{j''}^{(2)} \otimes u_{k''}^{(3)}$, By selecting the basis, we obtain an orthogonal projection to a lower-dimensional subspace.

and maximises the criterion

$$J_+ = \mathrm{E}\left(\|\hat{\mathcal{X}}_i\|_{\mathrm{F}}^2\right), \tag{2.98}$$

with respect to the conditions

$$\boldsymbol{U}^{(j)\top}\boldsymbol{U}^{(j)} = \boldsymbol{I}_j, \tag{2.99}$$

where $\boldsymbol{I}_j,\ j = 1, 2, \ldots, N$ are the identity matrices in $\mathbb{R}^{I_j \times I_j}$. By fixing $\{\boldsymbol{U}^{(j)}\}_{j=1}^N$ except $\boldsymbol{U}^{(j')}, j' \in \{1, 2, \ldots, N\}$, we have

$$J_j = \mathrm{E}\left(\|\boldsymbol{U}^{(j)\top}\mathcal{X}_{i,(j)}\mathcal{X}_{i,(j)}^\top\boldsymbol{U}^{(j)}\|_{\mathrm{F}}^2\right), \tag{2.100}$$

where $\mathcal{X}_{i,(j)},\ j = 1, 2, \ldots, N$, are the j-mode unfolded tensors of $\mathcal{X}_i$.

Eigendecomposition problems are derived by computing the extremals of

$$E_j \ = \ J_j + tr((\boldsymbol{I}_j - \boldsymbol{U}^{(j)\top}\boldsymbol{U}^{(j)})\boldsymbol{\Sigma}^{(j)}),\ j = 1, 2, \ldots, N. \tag{2.101}$$

For matrices $\boldsymbol{M}^{(j)} = \frac{1}{N}\sum_{i=1}^N \mathcal{X}_{i,(j)}\mathcal{X}_{i,(j)}^\top,\ j = 1, 2, \ldots, N$, the optimisation of J_- and J_+ derives the eigenvalue decomposition

$$\boldsymbol{M}^{(j)}\boldsymbol{U}^{(j)} = \boldsymbol{U}^{(j)}\boldsymbol{\Sigma}^{(j)}, \tag{2.102}$$

where $\boldsymbol{\Sigma}^{(j)} \in \mathbb{R}^{I_j \times I_j},\ j = 1, 2, \ldots, N$, are diagonal matrices satisfying the relationships $\sigma_k^{(j)} = \sigma_k^{(j')},\ k \in \{1, 2, \ldots, K\}$ for

$$\boldsymbol{\Sigma}^{(j)} \ = \ \mathrm{diag}(\lambda_1^{(j)}, \lambda_2^{(j)} \cdots, \lambda_K^{(j)}, 0 \cdots, 0). \tag{2.103}$$

The optimisation of each J_j derives the eigendecomposition problems in eq. (2.102). However, for the optimisation of $\{J_j\}_{j=1}^N$, there is no closed-form solution to this maximisation problem [100, 99]. Algorithm 1.3 is the iterative procedure of the TPCA for Nth-order tensors [112]. This algorithm is one of alternating-least-square (ALS) algorithms for tensors. In Algorithm 1.3 of K iteration, time complexity is $\mathcal{O}(KI_k^2)$, where $I_j \leq I_k,\ j \neq k$, due to the eigendecomposition problem.

For Algorithm 1.3, we have the following property.

Proposition 2.3 *The TPCA for Nth-order tensors without iteration in Algorithm 1.3 is equivalent to the HOSVD for Nth-order tensors if dimensions of a projected tensor are coincident to ones of each mode of an original tensor.*

For Nth-order tensors, there are $N!$ combinations in selecting the order of modes in a tensor-to-tensor projection for Algorithm 1.3. For the selection of combinations for Algorithm 1.3, we have the following property [112].

Algorithm 1.3: Iterative method in the MPCA for Nth-order tensors

Input: A set of tensors $\{\mathcal{X}_i\}_{i=1}^M$. Dimension of projected tensors $\{k_j\}_{j=1}^N$. A maximum number of iteration K. A sufficiently small number η.

Output: A set of projection matrices $\{\boldsymbol{U}^{(j)}\}_{j=1}^N$.

1: Compute the eigendecomposition of a covariant matrix $\boldsymbol{M}^{(j)} = \frac{1}{M}\sum_{i=1}^M \mathcal{X}_{i,(j)}\mathcal{X}_{i,(j)}^\top$, where $\mathcal{X}_{i,(j)}$ is an j-mode unfolded $\mathcal{X}_i$, for $j = 1, 2, \ldots, N$.

2: Construct projection matrices by selecting eigenvectors corresponding to the k_j largest eigenvalues for $j = 1, 2, \ldots, N$.

3: Compute $\Psi_0 = \sum_{i=1}^M \|\mathcal{X}_i \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \cdots \times_N \boldsymbol{U}^{(N)\top}\|_{\mathrm{F}}$.

4: Iteratively Compute the following procedure.

for $k = 1, 2, \ldots, K$

for $j = 1, 2, \ldots, N$

Update $\boldsymbol{U}^{(j)}$ by decomposing matrix $\sum_{i=1}^M \boldsymbol{W}_{i,(j)}^{(-j)}\boldsymbol{W}_{i,(j)}^{(-j)\top}$, where $\boldsymbol{W}_{i,(j)}^{(-j)}$ is an unfolding of $\mathcal{X}_i \times_1 \boldsymbol{U}^1 \cdots \times_{j-1} \boldsymbol{U}^{(j-1)} \times_{j+1} \boldsymbol{U}^{(j+1)} \cdots \times_N \boldsymbol{U}^{(N)}$ for a mode j.

end

Compute $\Psi_k = \sum_{i=1}^N \|\mathcal{X}_i \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \cdots \times_N \boldsymbol{U}^{(N)\top}\|_{\mathrm{F}}$

if $|\Psi_k - \Psi_{k-1}| < \eta$

break

end

Proposition 2.4 *For Nth-order tensors, the selection of order of modes does not effect to the results of a tensor-to-tensor projection since n-mode projection is cumulative.*

From these two properties, we adopt Algorithm 1.3 [112] to solve the optimisation of $\{J_j\}_{j=1}^N$. For a set of orthonormal vectors $\{\boldsymbol{e}_k\}_{k=1}^K$, $\boldsymbol{e}_i^\top \boldsymbol{e}_j = \delta_{ij}$, we set orthogonal projection matrices $\boldsymbol{P}^{(j)} = \sum_{k=1}^{k_j} \boldsymbol{e}_k\boldsymbol{e}_k^\top$ for $j = 1, 2, 3$. Using these $\{\boldsymbol{P}^{(j)}\}_{j=1}^N$, the low-rank tensor approximation [99] is achieved by

$$\mathcal{Y} = \mathcal{X} \times_1 (\boldsymbol{P}^{(1)}\boldsymbol{U}^{(1)}) \times_2 (\boldsymbol{P}^{(2)}\boldsymbol{U}^{(2)}) \cdots \times_N (\boldsymbol{P}^{(N)}\boldsymbol{U}^{(N)}), \tag{2.104}$$

where $\boldsymbol{P}^{(j)}$ selects k_j bases of orthogonal matrices $\boldsymbol{U}^{(j)}$. The low-rank approximation using eq. (2.104) is used for compression in the TPCA for Nth-order tensors.

For the HOSVD for Nth-order tensors, we have the following theorem.

Theorem 2.5 *The HOSVD method is equivalent to the vector PCA method in the compression of Nth-order tensors.*

(*Proof*) The equation

$$\mathcal{X} \times_1 (\boldsymbol{P}^{(1)}\boldsymbol{U}^{(1)})^\top \times_2 (\boldsymbol{P}^{(2)}\boldsymbol{U}^{(2)})^\top \cdots \times_N (\boldsymbol{P}^{(N)}\boldsymbol{U}^{(N)})^\top = \mathcal{Y} \tag{2.105}$$

is equivalent to

$$(\boldsymbol{P}^{(N)}\boldsymbol{U}^{(N)} \otimes \ldots \boldsymbol{P}^{(2)}\boldsymbol{U}^{(2)} \otimes \boldsymbol{P}^{(1)}\boldsymbol{U}^{(1)})^\top \mathrm{vec}\mathcal{X} = \mathrm{vec}\mathcal{Y}. \tag{2.106}$$

(Q.E.D.)

This theorem implies that N-dimensional discrete cosine transform (ND-DCT) is an acceptable approximation of the HOSVD for Nth-order tensors since this is the analogy of the approximation of the PCA of two-dimensional images by the 2DDCT [130, 78].

Furthermore, we have the following theorem.

Theorem 2.6 *The compression of Nth-order tensors computed by the HOSVD is equivalent to the compression computed by the TPCA.*

(*Proof*) The projection that selects $K = k_1 k_2 \ldots k_N$ bases of the tensor space spanned by $u^{(1)}_{i_1} \circ u^{(2)}_{i_2} \circ u^{(3)}_{i_3}$, $i_j = 1, 2, \ldots, k_j$ for $j = 1, 2, \ldots, N$, is

$$\begin{aligned}&(\boldsymbol{P}^{(N)}\boldsymbol{U}^{(N)} \otimes \ldots \boldsymbol{P}^{(2)}\boldsymbol{U}^{(2)} \otimes \boldsymbol{P}^{(1)}\boldsymbol{U}^{(1)}) \\ &\quad = (\boldsymbol{P}^{(N)} \otimes \ldots \boldsymbol{P}^{(2)} \otimes \boldsymbol{P}^{(1)})(\boldsymbol{U}^{(N)} \otimes \ldots \boldsymbol{U}^{(2)} \otimes \boldsymbol{U}^{(1)}) = \boldsymbol{P}\boldsymbol{W},\end{aligned} \tag{2.107}$$

where $\boldsymbol{W}$ and $\boldsymbol{P}$ are an orthogonal matrix and the orthogonal projection matrix, respectively. Therefore, HOSVD is equivalent to TPCA for third-order tensors.

2.3 Discrete Cosine Transform

2.3.1 One-Dimensional Discrete Cosine Transform

For a sampled one-dimensional signal represented by a vector, that is a first-order tensor, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^\top$, we have transformed signal $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^\top$ using the following four types of the discrete cosine transform (DCT) [5, 137].

DCT-I

$$\begin{aligned}&y_k = \tfrac{1}{2}(x_0 + (-1)^k)x_{(n-1)}) + \textstyle\sum_{j=1}^{n-2} x_j \cos\left(\frac{\pi}{n-1}jk\right), \\ &k = 0, 1, \ldots, n-1.\end{aligned} \tag{2.108}$$

DCT-II

$$y_k = \sum_{j=0}^{n-1} x_j \cos\left(\frac{\pi}{n}(j+\frac{1}{2})k\right), \quad k = 0, 1, \ldots, n-1. \tag{2.109}$$

DCT-III

$$y_k = \frac{1}{2} + \sum_{j=1}^{n-1} x_j \cos\left(\frac{\pi}{n}(j+\frac{1}{2})k\right), \quad k = 0, 1, \ldots, n-1. \tag{2.110}$$

DCT-IV

$$\begin{array}{l} y_k = \sum_{j=0}^{n-1} x_j \cos\left(\frac{\pi}{n}\left(j+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\right), \\ k = 0, 1, \ldots, n-1. \end{array} \tag{2.111}$$

An inverse transform of the DCT-I is the DCT-I scaled by $\frac{2}{(n-1)}$. The DCT-II and the DCT-III are transposes of one another. The DCT-III scaled by $\frac{2}{N}$ is an inverse transform of the DCT-II. An inverse transform of the DCT-IV is the DCT-IV scaled by $\frac{2}{n}$.

If we adopt DCT-II for a transform of $\boldsymbol{x}$, we have matrix representation of DCT as

$$\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x}, \quad \boldsymbol{D} = ((d_{ij})), \quad d_{ij} = \cos\left(\frac{\pi}{n}\left((j-1)+\frac{1}{2}\right)(i-1)\right), \tag{2.112}$$

where $i, j = 1, 2, \ldots n$. Furthermore, the inverse of the DCT is given by

$$\boldsymbol{x} = \boldsymbol{D}^\top \boldsymbol{y}. \tag{2.113}$$

For a integer $k \leq n$, we set an orthogonal projection matrix

$$\boldsymbol{P} = (p_{ij}), \quad p_{ij} = \begin{cases} 1 & \text{if } i = j \text{ and } i, j \leq k, \\ 0 & \text{otherwise.} \end{cases} \tag{2.114}$$

Using this orthogonal projection matrix, we define DCT-based reduction

$$\hat{\boldsymbol{x}} = (\boldsymbol{P}\boldsymbol{D})\boldsymbol{x}. \tag{2.115}$$

The reconstruction from $\hat{\boldsymbol{x}}$ is given by

$$\tilde{\boldsymbol{x}} = (\boldsymbol{P}\boldsymbol{D})^\top \hat{\boldsymbol{x}}. \tag{2.116}$$

The difference between $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ is reconstruction error

$$\epsilon = \|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_2. \tag{2.117}$$

The time complexity of the naive DCT is $\mathcal{O}(n^2)$. If we use fast Fourier transform (FFT), the time complexity is $\mathcal{O}(n \log n)$.

2.3.2 Two-Dimensional Discrete Cosine Transform

For sampled two-dimensional signal represented by a matrix, that a is second-order tensor, $\boldsymbol{X} \in \mathbb{R}^{m \times n}$, setting $\boldsymbol{D}_{\mathrm{L}} \in \mathbb{R}^{m \times m}$ and $\boldsymbol{D}_{\mathrm{R}} \in \mathbb{R}^{n \times n}$ to be DCT matrices defined in section 2.3.1, we have the two-dimensional discrete transform (2DDCT)

$$\boldsymbol{Y} = \boldsymbol{D}_{\mathrm{L}} \boldsymbol{X} \boldsymbol{D}_R^{\top}, \tag{2.118}$$

and its inverse transform

$$\boldsymbol{X} = \boldsymbol{D}_{\mathrm{L}}^{\top} \boldsymbol{Y} \boldsymbol{D}_R. \tag{2.119}$$

For integers $k_1 \leq m$ and $k_2 \leq n$, in the same manner of eq. (2.114), we have orthogonal projection matrices $\boldsymbol{P}_L$ of size $k_1 \times m$ and $\boldsymbol{P}_R$ of size $k_2 \times n$. Using $\boldsymbol{P}_L$ and $\boldsymbol{P}_R$, we define 2DDCT-based reduction

$$\hat{\boldsymbol{X}} = (\boldsymbol{P}_L \boldsymbol{D}_{\mathrm{L}}) \boldsymbol{X} (\boldsymbol{P}_R \boldsymbol{D}_R)^{\top}. \tag{2.120}$$

The reconstruction from $\hat{\boldsymbol{X}}$ is given by

$$\tilde{\boldsymbol{X}} = (\boldsymbol{P}_L \boldsymbol{D}_{\mathrm{L}})^{\top} \hat{\boldsymbol{X}} (\boldsymbol{P}_R \boldsymbol{D}_R). \tag{2.121}$$

The difference between $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$ is reconstruction error

$$\epsilon = \|\boldsymbol{X} - \tilde{\boldsymbol{X}}\|_F. \tag{2.122}$$

This 2DDCT-based reduction is an acceptable approximation of the compression by the PCA, MEV and 2DSVD.

The time complexity of the naive 2DDCT is $\mathcal{O}(n^3)$ for a matrix of size $n \times n$. If we use fast Fourier transform (FFT), the time complexity is $\mathcal{O}(n \log n)$.

2.3.3 Three-Dimensional Discrete Cosine Transform

For a three-dimensional signal represented by third-order tensor $\mathcal{X} \in \mathbb{R}^{m \times n \times l}$, setting DCT matrices $\boldsymbol{D}^{(1)} \in \mathbb{R}^{m \times m}$, $\boldsymbol{D}^{(2)} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{D}^{(3)} \in \mathbb{R}^{l \times l}$, we have the three-dimensional discrete cosine transform (3DDCT)

$$\mathcal{Y} = \mathcal{X} \times_1 \boldsymbol{D}^{(1)} \times_2 \boldsymbol{D}^{(2)} \times_3 \boldsymbol{D}^{(3)} \tag{2.123}$$

and its inverse transform

$$\mathcal{X} = \mathcal{Y} \times_1 \boldsymbol{D}^{(1)\top} \times_2 \boldsymbol{D}^{(2)\top} \times_3 \boldsymbol{D}^{(3)\top}. \tag{2.124}$$

For integers $k_1 \leq m$, $k_2 \leq n$ and $k_3 \leq l$, in the same manner of eq. (2.114), we have orthogonal projection matrices $\boldsymbol{P}^{(1)}$ of size $k_1 \times m$, $\boldsymbol{P}^{(2)}$ of

size $k_2 \times n$ and $\boldsymbol{P}^{(3)}$ of size $k_3 \times l$. Using $\boldsymbol{P}^{(1)}$, $\boldsymbol{P}^{(2)}$ and $\boldsymbol{P}^{(3)}$, we define 3DDCT-based reduction

$$\hat{\mathcal{X}} = \mathcal{X} \times_1 (\boldsymbol{P}^{(1)}\boldsymbol{D}^{(1)}) \times_2 (\boldsymbol{P}^{(2)}\boldsymbol{D}^{(2)}) \times_3 (\boldsymbol{P}^{(3)}\boldsymbol{D}^{(3)}) \tag{2.125}$$

The reconstruction from $\hat{\mathcal{X}}$ is given by

$$\tilde{\mathcal{X}} = \hat{\mathcal{X}} \times_1 (\boldsymbol{P}^{(1)}\boldsymbol{D}^{(1)})^\top \times_2 (\boldsymbol{P}^{(2)}\boldsymbol{D}^{(2)})^\top \times_3 (\boldsymbol{P}^{(3)}\boldsymbol{D}^{(3)})^\top. \tag{2.126}$$

The difference between $\mathcal{X}$ and $\hat{\mathcal{X}}$ is reconstruction error

$$\epsilon = \|\mathcal{X} - \tilde{\mathcal{X}}\|_{\mathrm{F}}. \tag{2.127}$$

The 3DDCT is an acceptable approximation of the compression by the PCA, MPCA and 3DSVD.

The time complexity of the naive 3DDCT is $\mathcal{O}(n^4)$ for a matrix of size $n \times n \times n$. If we use fast Fourier transform (FFT), the time complexity is $\mathcal{O}(n \log n)$.

2.3.4 N-Dimensional Discrete Cosine Transform

For a Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, setting DCT matrices $\boldsymbol{D}^k \in \mathrm{R}^{I_k \times I_k}$ for $k = 1, 2, \ldots, N$, we have N-dimensional discrete cosine transform (NDDCT)

$$\mathcal{Y} = \mathcal{X} \times_1 \boldsymbol{D}^{(1)} \times_2 \boldsymbol{D}^{(2)} \cdots \times_N \boldsymbol{D}^{(N)}. \tag{2.128}$$

and its inverse transform

$$\mathcal{X} = \mathcal{Y} \times_1 \boldsymbol{D}^{(1)\top} \times_2 \boldsymbol{D}^{(2)\top} \cdots \times_N \boldsymbol{D}^{(N)\top}. \tag{2.129}$$

For N integers $k_1 \leq I_1$, $k_2 \leq I_2$, $\ldots$, $k_N \leq I_N$, in the same manner of eq. (2.114), we have N orthogonal projection matrices $\boldsymbol{P}^{(1)}$ of size $k_1 \times I_1$, $\boldsymbol{P}^{(2)}$ of size $k_2 \times I_2$, $\ldots$, $\boldsymbol{P}^{(N)}$ of size $k_N \times I_N$. Using these N orthogonal projection matrices, we define NDDCT-based reduction

$$\hat{\mathcal{X}} = \mathcal{X} \times_1 (\boldsymbol{P}^{(1)}\boldsymbol{D}^{(1)}) \times_2 (\boldsymbol{P}^{(2)}\boldsymbol{D}^{(2)}) \cdots \times_N (\boldsymbol{P}^{(N)}\boldsymbol{D}^{(N)}). \tag{2.130}$$

The reconstruction from $\hat{\mathcal{X}}$ is given by

$$\tilde{\mathcal{X}} = \hat{\mathcal{X}} \times_1 (\boldsymbol{P}^{(1)}\boldsymbol{D}^{(1)})^\top \times_2 (\boldsymbol{P}^{(2)}\boldsymbol{D}^{(2)})^\top \cdots \times_N (\boldsymbol{P}^{(N)}\boldsymbol{D}^{(N)})^\top. \tag{2.131}$$

The difference between $\mathcal{X}$ and $\tilde{\mathcal{X}}$ is reconstruction error

$$\epsilon = \|\mathcal{X} - \tilde{\mathcal{X}}\|_{\mathrm{F}}. \tag{2.132}$$

This NDDCT is an acceptable approximation for the NDTPCA.

The time complexity of the naive NDDCT is $\mathcal{O}(n^N)$ for a matrix of size $n \times n \times n \cdots \times n$. If we use fast Fourier transform (FFT), the time complexity is $\mathcal{O}(n \log n)$.

2.3.5 Relation to Scale Space and Pyramid Transform

Planar Scale Space

In the two-dimensional Euclidean space $\mathbb{R}^2$, for an orthogonal coordinate system x-y defined in $\mathbb{R}^2$, a vector in $\mathbb{R}^2$ is expressed by $\boldsymbol{x} = (x, y)^\top$ where $\cdot^\top$ is the transpose of a vector. The solution of the linear diffusion equation

$$\frac{\partial}{\partial \tau} f(\boldsymbol{x}, \tau) = \Delta f(\boldsymbol{x}, \tau),\ \tau > 0,\ f(\boldsymbol{x}, 0) = f(\boldsymbol{x}) \tag{2.133}$$

with the boundary condition $\lim_{|\boldsymbol{x}| \to \infty} e^{\tau |\boldsymbol{x}|} f(\boldsymbol{x}) = 0$ defines the general image of the function $f(\boldsymbol{x})$ in the linear scale space. Setting $|\boldsymbol{x}|$ to be the length of $\boldsymbol{x}$, the solution of eq. (2.133) is obtained as

$$G *_2 f(\boldsymbol{x}) = f(\boldsymbol{x}, \tau) = \frac{1}{4\pi\tau} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\boldsymbol{y}) \exp(-\frac{|\boldsymbol{x} - \boldsymbol{y}|^2}{4\tau}) d\boldsymbol{y}. \tag{2.134}$$

The Hermite functions are eigenfunctions of the second order differential equation

$$\left(-\frac{\hbar}{2m^2} \frac{\partial^2}{\partial x^2} + \frac{1}{2} k x^2 \right) \phi(x) = E \phi(x) \tag{2.135}$$

The solution is

$$\phi_n(x, \omega) = A e^{-\frac{y^2}{2}} H_n(y),\ y = \sqrt{\frac{m\omega}{\hbar}} x,\ \omega = \sqrt{\frac{k}{m}} \tag{2.136}$$

Moreover, for the linear heat equation we have the relation

$$\left(\frac{\partial^2}{\partial x^2} - \frac{\partial}{\partial \tau} \right) \phi_n(x, \tau) = 0. \tag{2.137}$$

Therefore, a signal in linear scale space is expressed as

$$f(x, \tau) = \sum_{n=0}^{\infty} a_n \phi_n(x, \tau) \tag{2.138}$$

for

$$a_n = \int_{-\infty}^{\infty} f(x) \phi_n(x, \tau) dx \tag{2.139}$$

Setting $P_{mn}(x, y, \tau) = \phi_m(x, \tau) \phi_n(y, \tau)$ eq. (2.137) becomes

$$\left(\Delta - \frac{\partial}{\partial \tau} \right) P_{mn}(x, y, \tau) = 0. \tag{2.140}$$

Therefore, for functions on Euclidean plane, we have the relation

$$\begin{aligned} f(x,y,\sigma) &= \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} a_{mn}P_{mn}(x,y,\tau), \\ & a_{mn} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)P_{mn}(x,y,\tau)dxdy. \end{aligned} \tag{2.141}$$

Discrete Heat Equation

Here, we deal with numerical computation of the partial differential equation

$$\frac{\partial f}{\partial \tau} = \frac{\partial^2 f}{\partial x^2} \tag{2.142}$$

in $\mathbb{R}^2 \times \mathbb{R}_+$. We set a matrix

$$\boldsymbol{A} = \begin{pmatrix} -1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \end{pmatrix}, \tag{2.143}$$

where the size of $\boldsymbol{A}$ is $N \times N$. The eigenvalues of $\boldsymbol{A}$ is $4\sin\frac{\pi k}{2N}$ for $k = 0, 1, \ldots, n-1$ Furthermore, the eigenmatrix of $\boldsymbol{A}$ is

$$\begin{aligned} \boldsymbol{U}_N &= ((\epsilon \cos\frac{(2j+1)i}{2\pi N})) = ((u_{ij}^N)), \\ & \epsilon = \begin{cases} 1 & \text{if } j = 0 \\ \frac{1}{\sqrt{2}} & \text{otherwise.} \end{cases} \end{aligned} \tag{2.144}$$

Then, using semi-implicit and explicit discretisation, we have the form

$$\begin{aligned} \frac{\boldsymbol{f}^{(m+1)} - \boldsymbol{f}^{(m)}}{\tau} &= \frac{1}{2}\boldsymbol{A}\boldsymbol{f}^{(m+1)}, \\ \frac{\boldsymbol{f}^{(m+1)} - \boldsymbol{f}^{(m)}}{\tau} &= \frac{1}{2}\boldsymbol{A}\boldsymbol{f}^{(m)}. \end{aligned} \tag{2.145}$$

These equations are redescribed as

$$\boldsymbol{f}^{(n+1)} = (\boldsymbol{I} - \frac{\tau}{2}\boldsymbol{A})^{-1}\boldsymbol{f}^{(n)}, \ \boldsymbol{f}^{(n+1)} = (\boldsymbol{I} + \frac{\tau}{2}\boldsymbol{A})\boldsymbol{f}^{(n)}. \tag{2.146}$$

For the Neumann boundary condition, that is, the gradient of the solution normal to the boundary is zero, setting

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\top} \tag{2.147}$$

to be the orthogonal decomposition of matrix $\boldsymbol{A}$, that is

$$\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_0, \lambda_1, \cdots, \lambda_{N-1}). \tag{2.148}$$

and

$$\boldsymbol{U}_M = (\boldsymbol{u}_0^M, \boldsymbol{u}_1^M, \cdots, \boldsymbol{u}_{M-1}^M) = ((u_{ij}^M)). \tag{2.149}$$

for $\boldsymbol{U}_M^\top \boldsymbol{U}_M = \boldsymbol{I}$.

Eigenfunctions in Discrete Scale Space

Since the two-dimensional discrete Laplacian

$$\boldsymbol{L} = \boldsymbol{A} \oplus \boldsymbol{A} = \boldsymbol{A} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{A} \tag{2.150}$$

is decomposed as

$$\boldsymbol{L} = (\boldsymbol{U} \otimes \boldsymbol{U})(\boldsymbol{\Lambda} \oplus \boldsymbol{\Lambda})(\boldsymbol{U} \otimes \boldsymbol{U})^\top \tag{2.151}$$

the eigenfunctions for discrete scale space is $\mathcal{L}(\{\boldsymbol{u}_k\}_{k=0}^{M-1})$. Therefore, we have the following properties.

Proposition 2.5 *The discrete scale space filtering is a linear transform* $\mathcal{L}(\{\boldsymbol{u}_k\}_{k=0}^{M-1})$ *to* $\mathcal{L}(\boldsymbol{u}_k\}_{k=0}^{M-1})$.

Proposition 2.6 *The two-dimensional discrete scale space filtering is a linear transform* $\mathcal{L}(\{\boldsymbol{u}_i\boldsymbol{u}_j\}_{i=0,j=0}^{M-1,M-1})$ *to* $\mathcal{L}(\{\boldsymbol{u}_i\boldsymbol{u}_j\}_{i=0,j=0}^{M-1,M-1})$.

Pyramid transform of Signals

The pyramid transform

$$g_n = \frac{1}{4}(f_{2n-1} + 2f_{2n} + f_{2n+1}) \tag{2.152}$$

for the sequence $\{f_n\}_{n=-\infty}^{\infty}$ is redescribed as

$$g_n = h_{2n}, \quad h_n = \frac{1}{4}(f_{n-1} + 2f_n + f_{n+1}). \tag{2.153}$$

These relations imply that the pyramid transform is achieved by downsampling after moving average.

For a continuous function, downsampling after convolution is expressed as

$$g(x) = h(\sigma x, \tau) = \int_{-\infty}^{\infty} k_\tau(\sigma x - y) f(y) dy \tag{2.154}$$

$$h(x, \tau) = \int_{-\infty}^{\infty} k_\tau(x - y) f(y) dy. \tag{2.155}$$

If

$$k_\tau(x) = \frac{1}{\sqrt{2\pi\tau}} \exp(-\frac{x^2}{2\tau}), \tag{2.156}$$

$h(x)$ is the solution of

$$\frac{\partial h}{\partial \tau} = \frac{1}{2}\frac{\partial^2 h}{\partial x^2} \tag{2.157}$$

for $h(x,0) = f(x)$. Moreover, discretisation of the diffusion equation

$$h_i^{(n+1)} - h_i^{(n)} = \frac{1}{2}\left(\frac{h_{i+1}^{(n)} - 2h_i^{(n)} + h_{i-1}^{(n)}}{2}\right) \tag{2.158}$$

derives the discrete convolution

$$h_i = \frac{1}{4}h_{i+1} + \frac{1}{2}h_i + \frac{1}{4}h_{i-1}. \tag{2.159}$$

For the second order differential matrix $\boldsymbol{A}$, with the Neumann condition, setting

$$\boldsymbol{W} = \frac{1}{2}\boldsymbol{A} + 2\boldsymbol{I} = \begin{pmatrix} 3 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 2 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 2 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 & 3 \end{pmatrix}$$

we have the relation

$$\boldsymbol{h}^{(m+1)} = \boldsymbol{W}\boldsymbol{h}^{(m)} = \left(\frac{1}{2}\boldsymbol{A} + 2\boldsymbol{I}\right)\boldsymbol{h}^{(m)} \tag{2.160}$$

as the matrix expression of the eq. (2.159). The eigenvalue of $\boldsymbol{W}$ is $\frac{1}{2}\lambda_i + 2$ where λ_i is a eigenvalue of $\boldsymbol{A}$. Furthermore, the eigenmatrix of $\boldsymbol{W}$ is the that of $\boldsymbol{A}$. Moreover, eq. (2.159) computes the weighted average of $h_i^{(n)}$. Therefore, the eigenfunction of the second order differentiation and the average operation coincide each other.

Eigenfunction of Pyramid Transform

Using the matrix $\boldsymbol{A}$, the scale space evolution and smoothing of discrete signals are expressed as

$$\boldsymbol{f}^{(m+1)} = \left(\boldsymbol{I} - \tau\frac{1}{2}\boldsymbol{A}\right)^{-1}\boldsymbol{f}^{(m)} \tag{2.161}$$

$$\boldsymbol{f}^{(m+1)} = \left(\frac{1}{2}\boldsymbol{A} + 2\boldsymbol{1}\right)\boldsymbol{f}^{(m)} \tag{2.162}$$

With the Neumann boundary condition, $\boldsymbol{A}$ is decomposed as

$$\boldsymbol{A} = \boldsymbol{U}_{2^n}^{\top} \boldsymbol{\Lambda} \boldsymbol{U}_{2^n} \tag{2.163}$$

where $\boldsymbol{U}_n$ is the DCT-II matrix of order 2^n. Therefore, setting

$$\boldsymbol{U}_{2^n} = (\boldsymbol{u}_0, \cdots, \boldsymbol{u}_{2^n-1}), \tag{2.164}$$

$\boldsymbol{u}_k$ is eigenvector of $\boldsymbol{A}$ associated to the eigenvalue λ_k. By applying downsampling to $\boldsymbol{f}^{(m)}$ with the factor 2, we have vector $\boldsymbol{f}_2^{(m)}$ in 2^{n-1} dimensional Euclidean space. Furthermore, the eigenvectors in this 2^{n-1} dimensional space are $\{u_k\}_{k=0}^{2^{n-1}}$ if we set $M = 2^n - 1$. Using these properties of eigenvectors, we have the following properties.

Proposition 2.7 *The Gaussian pyramid transform for signals is a linear transform from* $\mathcal{L}(\{\boldsymbol{u}_k\}_{k=0}^{2^n-1})$ *to* $\mathcal{L}(\{\boldsymbol{u}_k\}_{k=0}^{2^{n-1}-1})$.

Chapter 3

Recognition of Bilinear Forms

This Chapter is based on Publication of Journal Paper "2. Dimension Reduction and Construction of Feature Space for Image Pattern Recognition".

3.1 Dimension Reduction Methods for Image Pattern Recognition

This chapter focuses on the image pattern recognition that identifies particular instances of objects using extracted and normalised distributions of intensities of images.

In image pattern recognition, images are sampled and they can be embedded in a vector space. Image pattern recognition methods use a metric defined in a vector space. Dimension reduction reduces the dimensions of the feature space where classification and categorisation are performed. This procedure reduces the dimension of the feature space without significant loss of the recognition rate [114], while compression is the reduction of a dataset with a small reconstruction error [30] for the original data. In practice, as shown in Fig. 3.1, two types of methods are used for dimension reduction. One type reduces the dimension of the data in a sampled image space using image compression methods such as the pyramid transform (PT) and low-pass filtering. The other involves data compression in a vector space after vectorisation of sampled image patterns using operations such as the random projection (RP). The reduction and vectorisation operations are generally noncommutative as shown in Fig. 3.1. Table 3.1 summarises the abbreviations of the dimension-reduction methods and classifiers.

Nonexpansive mapping and topology-preserving mapping are the two fundamental dimension-reduction methods in image pattern recognition. Nonexpansive mapping is a Lipschitzian map if its Lipschitzian constant is less than

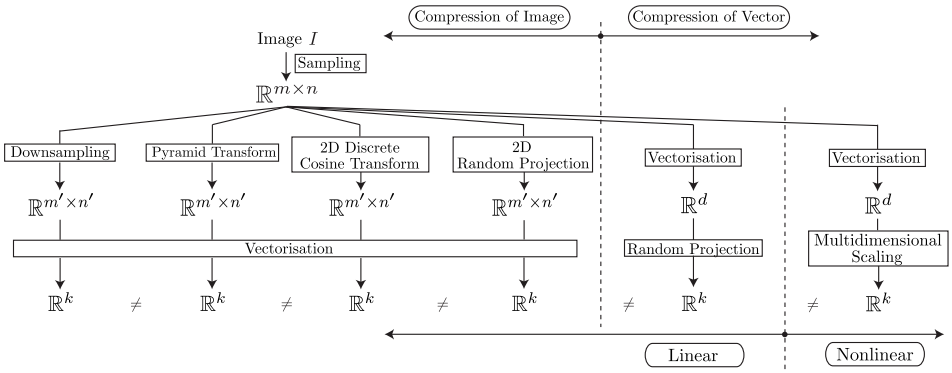


Figure 3.1: Differences in the dimension-reduction path among downsampling, the pyramid transform, the two-dimensional discrete cosine transformation, the two-dimensional random projection, the random projection and multidimensional scaling. After the sampling of an original image, dimension-reduction methods mainly follow two paths. In the first path, after the reduction of the image, it is converted to a vector. In the second path, after vectorisation, the dimension of the feature vector is reduced. Here, $m, m', n, n', d, k \in \mathbb{Z}$ and $n' < n, m' < m, k < d$.

or equal to one [85]. In the feature space, nonexpansive mapping can contract the distance and angle between points. This property causes changes in the similarity between patterns. Therefore, when we combine nonexpansive mapping, such as the PT, and other recognition methods, we cannot determine which procedure contributes to high and low recognition rates. This implies that if the PT is used as preprocessing for pattern classification, inner-product-based recognition can fail. On the other hand, topology-preserving mapping preserves the topology of the original feature space. Therefore, using topology-preserving mapping as a dimension-reduction method, we can evaluate pattern recognition techniques strictly.

The first contribution of this chapter is a mathematical analysis of dimension-reduction methods such that the PT, global two-dimensional discrete cosine transform (2DDCT) [1], the RP and two-dimensional random projection (2DRP). Although the PT is a well-known and widely used method of preprocessing in many applications, its property as a nonexpansive mapping for a feature space has not been presented. We reveal this property of the PT by comparing it with the RP and 2DRP, which are topology-

[1]In this chapter, the 2DDCT is applied to an image without partitioning while the JPEG and MPEG compression algorithms divide an image into blocks of 8×8 pixels before applying the 2DDCT.

Table 3.1: Glossary of abbreviations.

PT	Pyramid transform
PCA	Principal component analysis
2DSVD	Two-dimensional singular value decomposition
GPCA	Generalised principal component analysis
MEV	Marginal eingenvector
HOSVD	Higher-order singular value decomposition
TPCA	Tensor principal component analysis
MPCA	Multilinear principal component analysis
RP	Random projection
2DRP	Two-dimensional random projection
2DDCT	Two-dimensional discrete cosine transform
MDS	Multidimensional scaling
SM	Subspace method
MSM	Mutual subspace method
CMSM	Constraint mutual subspace method
2DTSM	Two-dimensional tensor subspace method

preserving mappings. We also derive the geometry-preserving property of the topology-preserving mapping. Furthermore, we clarify the relations among and the 2DDCT, principal component analysis (PCA), two-dimensional singular value decomposition (2DSVD) and tensor principal component analysis (TPCA). These relations imply that the 2DDCT provides an acceptable approximation for the PCA, the TPCA and the 2DSVD in pattern recognition as a fast and efficient computational method.

The second contribution is an experimental validation of these mathematically analysed methods. To validate the effects of dimension-reduction methods, we evaluate the recognition rate for eight datasets. For the recognition, we adopted subspace method (SM), mutual subspace method (MSM), constraint mutual subspace method (CMSM) and two-dimensional tensor subspace method (2DTSM). We tested pairs of these dimension-reduction techniques and classifiers for face recognition, spatial object recognition and character recognition. The results of the experiments show that both topology- and geometry-preserving properties are essential for dimension reduction since we cannot know the distribution of an image pattern in a feature space as *a priori*.

3.2 Related Works

Burt and Adelson [30] proposed a Laplacian pyramid for image encoding in which local operators of many scales but an identical shape serve as the basis function. Borgefors *et al.* [22] proposed a multivalued pyramid that preserves the geometrical and topological properties of images. They used a multivalued pyramid to construct a multiresolution skeleton that is useful in many image analysis tasks. Kropatsch *et al.* [96] proposed three pyramid methods based on graph theory. They showed both theoretically and experimentally that the number of vertices can be reduced. These pyramid methods preserve the geometrical properties of an image.

Second-order TPCA, which directly decomposes an image matrix, is used for two-dimensional images [111] as an extension of the PCA. According to a review on multilinear subspace learning [111], there are three basic projections for a tensor. The second-order TPCA uses a tensor-to-tensor projection consisting of 1- and 2-mode projections that act on columns and rows of images, respectively. Yang *et al.* [184] proposed two-dimensional principal component analysis (2DPCA) for image representation . Otsu [131] developed the marginal eigenvector (MEV) method, which is based on both 1- and 2-mode projections. Aase *et al.* [1] developed a singular value decomposition (SVD)-based image coding system. Ding and Ye [44] developed the 2DSVD, which is equivalent to the coding system of Aase *et al.*, as an extension of the SVD for image compression. The 2DSVD is also based on both 1- and 2-mode projections. The projections in the MEV method and 2DSVD are equivalent to the tensor-to-tensor projection for a second-order tensor. This mathematical property implies that the 2DSVD is a special case of the TPCA.

Ye *et al.* [185] proposed generalised principal component analysis (GPCA) for image compression that finds both 1- and 2-mode projections. The GPCA is a two-dimensional version of the iterative algorithm for the SVD [125]. Furthermore, the GPCA is a second-order version of the multilinear principal component analysis (MPCA) [112], which is an practical computation method of the TPCA. Moreover, in the MPCA, the projections obtained by higher-order singular value decomposition (HOSVD) [100] is used as the initial projections of the iterative algorithm. The iterative algorithms in the GPCA, MPCA and HOSVD are called alternating least squares algorithm. Independently from the MPCA, tensor rank-one decomposition (TROD) [172] was proposed for multidimensional data compression. The TROD is also an iterative algorithm based on the alternating least squares algorithm and the HOSVD. The minimisation problem in the TROD is coincident with that of the MPCA. A difference between the MPCA and TROD is that

the TROD finds rank-one tesnors as bases for a tensor subspace, while the MPCA find bases for each mode of tensors. As the extensions of the MPCA, by adding uncorrelation and sparsity constraints to the minimisation problem of decomposition in the MPCA, uncorrelated MPCA [113] and sparse higher-order PCA [8] were proposed, respectively.

Johnson and Lindenstrauss [83] showed the possibility of the low-distortion embedding of points from a high-dimensional Euclidean space to a low-dimensional Euclidean space. This low-distortion embedding method is called the RP. The RP approximately preserves the topology and geometry in a Euclidean space after the embedding. This property of the RP is useful for the dimension-reduction step in pattern recognition. Therefore, the RP has been used for various applications [13, 3, 143, 14]. Arya *et al.* [13] used the RP to approximately obtain the nearest neighbour. Achlioptas and McSherry [3] used the RP for PCA. Sakai and Imiya used the RP for clustering [143]. Baraniuk and Wakin [14] used the RP for manifold learning. Bingham and Mannila [19] showed the validity of the RP for the dimension reduction of noiseless images, noisy images and text data. In ref. [19], experimental results show that the RP preserves distances among the original high-dimensional data in a low-dimensional subspace.

Achlioptas [2] proposed an efficient RP using a sparse matrix. Watanabe *et al.* [178] suggested that the entries of a random matrix must be at least 4-wise independent to approximately preserve the pairwise distances. The Johnson-Lindenstrauss lemma can be proven even if four or more arbitrary entries in every row of a random matrix are statistically independent. Matousek [122] introduced a simple and self-contained proof of the Johnson-Lindenstrauss lemma that subsumes the basic and efficient version of the RP. Ailon and Liberty [6] proposed the fast Johnson-Lindenstrauss transform (FJLT) for the projection by a sparse random matrix. Sakai and Imiya [143] also proposed an improved normalised random linear map based on a dense random matrix as an efficient RP. They experimentally showed that the distortions of relative errors by this efficient RP are almost coincident with those by the RP.

Kernel methods are nonlinear dimension-reduction procedures. As a nonlinear dimension-reduction method, Schölkopf *et al.* [149] mapped data into a high-dimensional space by kernel trick. These projected data can be separated by hyperplanes, although they cannot be separated by hyperplanes in the original space. Therefore, in this high-dimensional space, we can use the PCA as kernel PCA. Multidimensional scaling (MDS) is a traditional method for information visualisation [21], which requires a mapping from high-dimensional feature space to two- or three-dimensional feature space. The classical MDS is computed by the PCA. The classical MDS is extended

by adopting a cost function for distances before and after mapping. This extended MDS is called the metric MDS. In the cost function of the metric MDS, a nonlinear mapping such as Sammon's mapping is used [21]. Sammon's mapping gives small distances a larger weight in the cost function. Using pairwise distances transformed by a nonlinear mapping, the metric MDS is practically computed by the PCA in the same way as in the classical MDS. Williams [179] clarified the relation between the metric MDS and the kernel PCA. He showed that the kernel PCA can be interpreted as a metric multidimensional scaling if the kernel is an isotropic function.

As another extension of the classical MDS, Tenenbaum *et al.* [157] developed the isomap. The difference of between the classical MDS and the isomap is the adoption of geodesic distances among data instead of the Euclidean distance. Roweis and Saul [140] developed local linear embedding (LLE). The LLE is uses the local linearity of a data manifold to find a weight-based representation of each point with its neighbours similarly to the metric MDS. Venna and Kaski [167] presented a comparison of the PCA, the metric MDS, the isomap and the LLE in the context of information visualisation. Vidal *et al.* [168] proposed the generalised principal component analysis. This generalised principal component analysis [2] gives a segmentation of a subspace spanned by the data of a category in a Euclidean space by finding multiple linear low-dimensional subspaces. Goh and Vidal [61] extended the standard LLE to deal with multiple submanifolds of a Riemannian space by introducing various Riemannian geometries, such as the Karcher mean, tangent space and geodesics. Harandi *et al.* [68] developed a method of geometry-aware dimension reduction for symmetric positive definite matrices. Their method learns a mapping from a high-dimensional symmetric positive definite manifold to a lower-dimensional one without relying on the tangent space approximations of the manifold.

Classification methods are mainly categorised in linear classification and nonlinear classification methods. There are two types of Linear classification methods: linear-function-based classification and subspace-based classification. Fisher [54] proposed linear discriminant analysis (LDA) for two-category classification. In the LDA, data are projected to a one-dimensional space by a linear projection and classified with a criterion. Vapnik and Lerner [164] proposed the support vector machine (SVM) for two-category classification. The SVM finds a hyperplane that separates two categories with the maximum margin between the hyperplane and data. The LDA and the SVM are linear-function-based classification methods that give a half-space

[2]This generalised principal component analysis is a different method to the GPCA [185] although they are the same name.

for two categories by deciding a hyperplane. Iijima [76] and Watanabe [175] introduced the PCA for the linear approximation of subspaces of multiple-category data. The PCA selects the subspace in which the covariance of class data is maximised. This method is called the SM. Furthermore, Iijima [76] introduced the constant normalisation of subspaces to the problem of character recognition. The constant normalisation in the PCA subtracts a constant bias since each image pattern contains a constant bias. The classical SM computes the orthogonal projection of inputs to each category. Itoh *et al.* proposed the 2DTSM as an extension of the SM for tensor pattern recognition [81]. Using the MEV to construct a tensor subspace for each class, the 2DTSM measures the distance between the input image and each class subspace.

As another extension of the SM, Maeda [116] proposed the MSM, which computes the orthogonal projection of subspaces spanned by inputs with perturbations. Fukui and Maki [57] proposed a combination of a generalisation of constant normalisation and the MSM. This method subtracts the elements in the common linear subspace of many categories. In the MSM, classification is based on angles among subspaces of each category. Cock and Moor [36] introduced the notion of subspace angles by considering the principal angles between two subspaces. Hamm and Lee [65] proposed a framework for Grassmann manifold learning. A Grassmann manifold is the set of fixed-dimensional linear subspaces in a Euclidean space. This learning framework unifies the view of the subspace-based learning method by formulating problems on a Grassmann manifold. The Grassmann manifold manipulate each subspace as a point in the Grassmann space, and feature extraction and classification are performed in the same space.

As a nonlinear method, Boser *et al.* [23] extended the linear SVM to a nonlinear SVM by applying the kernel trick with the sigmoid function to the linear SVM. This nonlinear SVM computes the hyperplane that divides data into two categories in a high-dimensional space. Furthermore, for data that cannot be divided with a linear function in a high-dimensional space, Cortes and Vapnik [39] developed a soft margin algorithm. This algorithm finds a hyperplane allowing the minimum number of outliers that exist in a margin. For multicategory classification, several methods that combine the nonlinear SVMs have been proposed [74].

In the image categorisation, the bag-of-visual words (BoW) method [153] is a common approach. In the BoW method, transformation-invariant local features, which are extracted from learning images, are used for the generation of a codebook. This codebook is a set of representative extracted local features. The BoW method assumes that an image is a set of local features. Therefore, images are represented as histograms that represent the frequen-

cies of occurrence of visual words in the codebook. In spite of constructing an order structure based on the histogram, vectors, which represent histograms, are embedded in a metric space. Csurka *et al.* [40] applied the BoW method for visual categorisation, which is the problem of identifying the object content of natural images while generalising across variations inherent to the object class. Sivic and Zisserman [152] applied the BoW method for object and image retrievals Generally, such categorisation requires an order structure. However, in visual categorisation, the distance between vectors that represent histograms is adopted in spite of using the order with respect to occurrences. In the dimension-reduction method for the BoW method, the PCA is used [82].

3.3 Mathematical Preliminaries

3.3.1 Pyramid Transform

We define the downsampling operation D_σ with factor σ as

$$g(x,y) = D_\sigma f(x,y) = f(\sigma x, \sigma y). \tag{3.1}$$

The upsampling operation U_τ of the factor is defined as

$$f(x,y) = U_\tau g(x,y) = \frac{1}{\tau^2} g(x/\tau, y/\tau). \tag{3.2}$$

For the downsampling and upsampling, we have the following properties.

Property 3.1 *These operations satisfy the relations*

$$\int_{\mathrm{R}^2} U_\sigma g(x,y) f(x,y) dxdy = \int_{\mathrm{R}^2} g(x,y) D_\sigma f(x,y) dxdy, \tag{3.3}$$

$$\int_{\mathrm{R}^2} |g(x,y)|^2 dxdy = \int_{\mathrm{R}^2} |U_\sigma g(x,y)|^2 dxdy \tag{3.4}$$

and

$$D_\sigma U_\sigma g(x,y) = g(x,y). \tag{3.5}$$

Therefore, $U_\sigma = D_\sigma^*$ is a partial isometric operator.

The PT is defined by a pair of a linear dimension-reduction operator R and its dual operator E. The linear dimension-reduction operator R is defined as

$$\begin{aligned} g(x,y) &= Rf(x,y), \\ &= \int\int_{\mathbf{R}^2} w_\sigma(u) w_\sigma(v) f(\sigma x - u, \sigma y - v) du dv, \end{aligned} \tag{3.6}$$

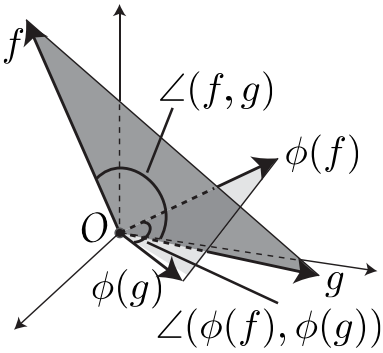


Figure 3.2: Angle between two functions and a nonexpansive map. $f, g \in H$ are functions and ϕ is a nonexpansive mapping. Here, $\angle(f, g)$ represents the angle between f and g.

$$w_\sigma(x) = \begin{cases} \frac{1}{\sigma}(1 - \frac{|x|}{\sigma}), & |x| \leq \sigma \\ 0, & |x| > \sigma \end{cases} . \tag{3.7}$$

The dual operation of R is

$$Eg(x, y) = \sigma^2 \int \int_{\mathbf{R}^2} w_\sigma(u) w_\sigma(v) g(\frac{x-u}{\sigma}, \frac{y-v}{\sigma}) du dv. \tag{3.8}$$

From Eq. (1), dimension reduction is achieved by downsampling a smoothed function with convolution kernel w_σ. In practical applications, $\sigma = 2$ is selected.

For a nonexpansive mapping ϕ such that

$$\|\phi(f) - \phi(g)\|_2 \leq r\|f - g\|_2,\ 0 \leq r \leq 1, \tag{3.9}$$

with the condition $\phi(f) \neq \lambda f$, we have the following theorem illustrated in Fig. 3.2.

Lemma 3.1 *Setting $\angle(f, g)$ to be the angle between f and g in the Hilbert space H, the relation*

$$\angle(\phi(f), \phi(g)) \leq \angle(f, g) \tag{3.10}$$

is satisfied.
(Proof) From the assumptions for the norms, we have the relations $\|\phi(f)\|_2 \leq \|f\|_2$, $\|\phi(g)\|_2 \leq \|g\|_2$ and $\|\phi(f) - \phi(g)\|_2 \leq \|f - g\|_2$. Furthermore, $\phi(f) \neq \lambda f$, $\phi(g) \neq \mu g$ and $\phi(f - g) \neq \nu(f - g)$. These relations imply the relation

$$\frac{(f, g)}{\|f\|_2 \|g\|_2} \leq \frac{(\phi(f), \phi(g))}{\|\phi(f)\|_2 \|\phi(g)\|_2}. \tag{3.11}$$

Setting $\theta = \angle(f, g)$, $\cos\theta = \frac{(f,g)}{\|f\|_2\|g\|_2}$. *Therefore, we have* $\angle(\phi(f), \phi(g)) = \theta_\phi \leq \theta$.

(Q.E.D.)

For L_1 and L_2, the norms are defined as

$$\|f\|_2 = \left(\int\int_{\mathrm{R}^2} |f(x,y)|^2 \, dxdy \right)^{\frac{1}{2}}, \tag{3.12}$$

$$\|g\|_1 = \int\int_{\mathrm{R}^2} |g(x,y)| \, dxdy. \tag{3.13}$$

For the convolution $h(x,y) = g(x,y) * f(x,y)$ of $f(x,y)$ and $g(x,y)$, we have the following proposition.

Proposition 3.1 *For the energy of a convolution, we have the following property:*

$$\|h(x,y)\|_2^2 = \|g(x,y) * f(x,y)\|_2^2 \leq \|g\|_1^2 \|f\|_2^2. \tag{3.14}$$

For a linear operator R, if $Rf \neq 0$, we have the following lemma.

Lemma 3.2 *For all* $f \in L_2$ *and* $g \in L_2$, *the relation*

$$\|Rf - Rg\|_2 \leq \frac{1}{\sigma}\|f - g\|_2 \tag{3.15}$$

is satisfied.

Lemma 3.3 *The PT is a linear nonexpansive and dimension reduction mapping.*

(*Proof*) From the definition, for two two-dimensional images, the linear operation in the PT reduces the size of the images to one quarter of their original size. Therefore, this operation is a linear dimension-reduction mapping. From Lemma 3.2, the PT is a nonexpansive mapping.

(Q.E.D.)

From Lemmas 3.1, 3.2 and 3.3, image reduction by the PT reduces the distance and angle between two images. Since it is a contraction map, in a practical scenario, the PT provides us with the same or greater similarity even for dissimilar patterns. From this property, we derive the following property.

Property 3.2 *For patterns whose similarity is originally large, Lemma 3.1 guarantees that the PT provides greater similarity than that of the original patterns. However, even for patterns whose similarity is originally small, the PT provides a greater similarity than the original one. This property can cause false-positive recognition.*

For a sampled function $f_{ij} = f(i, j)$, the downsampling D_σ and its dual operation U_τ are expressed as

$$D_\sigma f_{mn} = f_{\sigma m, \sigma n}, \tag{3.16}$$

$$U_\tau f_{mn} = \frac{1}{\tau^2} f_{m/\tau, n/\tau}. \tag{3.17}$$

Furthermore, the PT R and its dual transform E [30] are expressed as

$$Rf_{mn} = \sum_{i,j=-1}^{1} w_i w_j f_{2m-i,\, 2n-j}, \tag{3.18}$$

$$Ef_{mn} = 4 \sum_{i,j=-2}^{2} w_i w_j f_{\frac{m-i}{2}, \frac{n-j}{2}}, \tag{3.19}$$

where $w_{\pm 1} = \frac{1}{4}$ and $w_0 = \frac{1}{2}$. Moreover, the summation is carried out for integers $(m - i)$ and $(n - j)$. These two operations involve the reduction and expansion of the image size. As a nonexpansive mapping, the PT compresses the nth-order tensor with $\mathcal{O}(1/2^n)$, preserving the differential geometric structure of the tensor data.

3.3.2 Random Projection

The RP is a metric-embedding method that approximately preserves distances between points in the original space [166]. Furthermore, for an arbitrary set of points, the RP preserves angles among points, the volumes of simplexes [118], the lengths of smooth curves [4] and manifolds [14]. Figures 3.3(a), 3.3(b) and 3.3(c) respectively show the preservation of distances, angles and volumes, and manifolds by the RP.

For a set $X = \{\boldsymbol{x}_i\}_{i=1}^N$ of N points in d-dimensional Euclidean space, consider a mapping onto a set $\hat{X} = \{\hat{\boldsymbol{x}}_i\}_{i=1}^N$ in k-dimensional Euclidean space. For a vector $\boldsymbol{x} = (x_1, \ldots, x_d)^\top$, we define the Euclidean norm as $\|\boldsymbol{x}\|_2 = \left(\sum_{i=1}^d |x_i|^2\right)^{1/2}$. The Johnson-Lindenstrauss lemma indicates that there is a mapping that approximately preserves the distance between two arbitrary points [83].

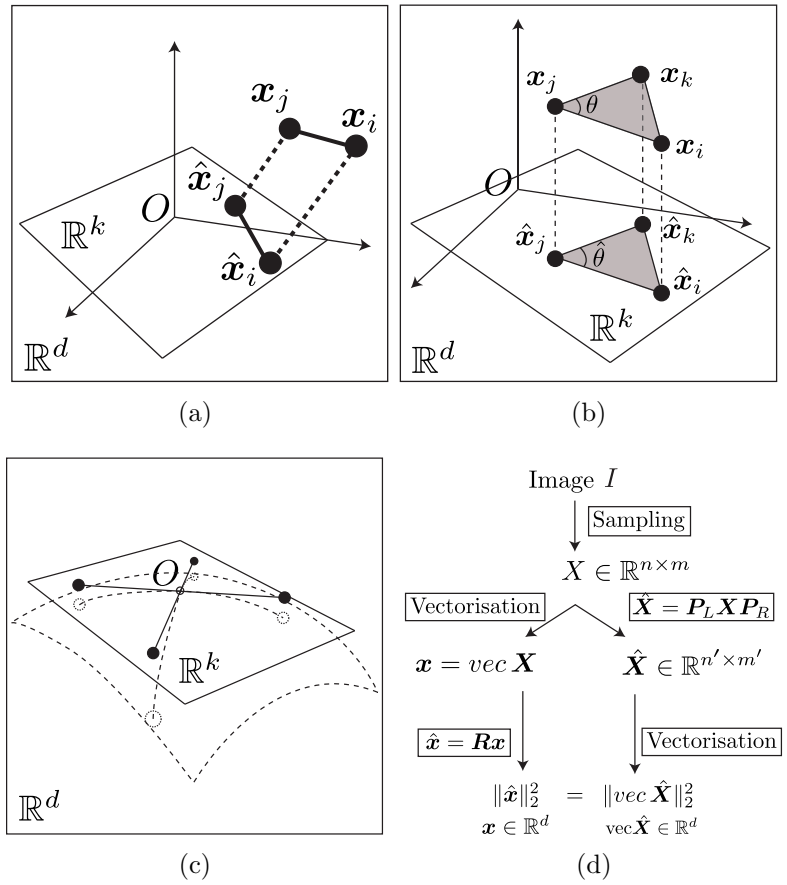


Figure 3.3: (a) Random projection (RP). Let $\boldsymbol{x}_i \in X$ be a point and $\hat{\boldsymbol{x}}_i = \boldsymbol{R}\boldsymbol{x}_i$. The distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is preserved in the projected space $\mathbb{R}^k$. (b) Preservation of angles and volumes. Points $\boldsymbol{x}_i, \boldsymbol{x}_j$ and $\boldsymbol{x}_k$ are in the original space, and points $\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j$ and $\hat{\boldsymbol{x}}_k$ are in the projected space. The RP preserves the angle θ in the projected space $\mathbb{R}^k$ as $\hat{\theta}$. The grey regions illustrate the area of a triangle. The RP also preserves areas and volumes. (c) Preservation of manifolds. The curved plane with dashed lines illustrates a manifold in the original space $\mathbb{R}^d$. The solid lines illustrate the projected manifold in $\mathbb{R}^d$. (d) Differences in two RP paths.

Property 3.3 *[83] (Johnson-Lindenstrauss lemma). For a subspace with dimension* $k \geq k_0 = \frac{9 \log N}{\epsilon^2 - \frac{2}{3}\epsilon^3} + 1 = \mathcal{O}(\epsilon^{-2} \log N)$*, where* ϵ *is a real number such that* $0 < \epsilon < \frac{1}{2}$*, a set* X *of* N *d-dimensional points* $\{\boldsymbol{x}_i\}_{i=1}^N$ *and an integer* k *with* $k \ll d$*, there exists a mapping* f *from* $\mathbb{R}^d$ *to* $\mathbb{R}^k$ *such that*

$$(1-\epsilon)\|\boldsymbol{x}_j - \boldsymbol{x}_i\|_2^2 \leq \|\hat{\boldsymbol{x}}_j - \hat{\boldsymbol{x}}_i\|_2^2 \leq (1+\epsilon)\|\boldsymbol{x}_j - \boldsymbol{x}_i\|_2^2, \tag{3.20}$$

for all $i, j = 1, 2, \ldots, N$.

From the viewpoint of pattern recognition, Property 3.3 indicates that this mapping approximately preserves the similarity between patterns regardless of the type of pattern or its distribution, although the mapping cannot preserve the geometry of an image.

For a set $X = \{\boldsymbol{x}_i\}_{i=1}^N$ of N points in d-dimensional Euclidean space such that $N \leq d$, let $\boldsymbol{R}$ be the $k \times d$ orthonormal matrix

$$\begin{pmatrix} \boldsymbol{r}_1^\top \\ \vdots \\ \boldsymbol{r}_k^\top \end{pmatrix} = \boldsymbol{R} \tag{3.21}$$

whose k row vectors $\{\boldsymbol{r}_i\}_{i=1}^k$ span a k-dimensional linear subspace in $\mathbb{R}^d$ ($k < d$). Multiplying by the uniform random orthonormal matrix $\boldsymbol{R}$, we obtain a low-dimensional representation $\hat{\boldsymbol{x}}_i$ for each $\boldsymbol{x}_i \in X$ as

$$\hat{\boldsymbol{x}}_i = \sqrt{\frac{d}{k}} \boldsymbol{R} \boldsymbol{x}_i, \tag{3.22}$$

which satisfies Property 3.3. We call this mapping in Eq. (3.22) the RP [166]. Figure 3.3(a) shows the basic idea of the RP. The scaling factor $\sqrt{\frac{d}{k}}$ is chosen to make the expected squared length of $\hat{\boldsymbol{x}}$ equal to the squared length of $\boldsymbol{x}$. That is, the RP satisfies

$$\mathrm{E}(\|\hat{\boldsymbol{x}}_i\|_2^2) = \|\boldsymbol{x}_i\|_2^2. \tag{3.23}$$

For the RP, we have the following Lemma.

Lemma 3.4 *The RP is a linear and topology-preserving dimension-reduction mapping.*

(*Proof*) According to Property 3.3, linear mappings of the RP approximately preserve the topology of the original space in a low-dimensional space.

(Q.E.D.)

There are various choices for the random matrix $\boldsymbol{R}$. The matrix of i.i.d. normal random variables with mean zero and variance $1/k$, $\mathcal{N}(0, 1/k)$, satisfies Lemma 3.3 [166]. The basis vectors $\boldsymbol{r}_i^\top, i = 1, \ldots, k$, of $\boldsymbol{R}$ are orthonormal in the sense of expectation as $\frac{k}{d}\,\mathrm{E}\,(\|\boldsymbol{r}_i\|_2^2) = 1,\ \mathrm{E}\left(\boldsymbol{r}_i^\top \boldsymbol{r}_j\right) = 0\ (i \neq j)$. A lower bound for k of

$$k_0 = \frac{4}{\epsilon^2/2 - \epsilon^3/3} \log N \tag{3.24}$$

is given in [42]. This lower bound shows that the dimension k of a low-dimensional space is independent of original dimension d. The lower bound k_0 is derived from the Markov inequality, and the actual approximation errors are much smaller than ϵ in most practical cases [143]. For an $\mathcal{N}(0, 1/k)$-based normalised random linear map, we have the following lemma [166]. Note that the scaling factor is different from that in Eq. (3.22) in a normal-distribution-based RP.

Lemma 3.5 *[166] Let each entry of an $n \times k$ matrix $\boldsymbol{R}$ be chosen independently from $\mathcal{N}(0, 1)$. Let $\hat{\boldsymbol{x}} = \frac{1}{\sqrt{k}}\boldsymbol{R}\boldsymbol{x}$ be a normalised random linear map for $\boldsymbol{x} \in \mathbb{R}^d$. Then for any $\epsilon > 0$,*

1. $\mathrm{E}(\|\hat{\boldsymbol{x}}\|_2^2) = \|\boldsymbol{x}\|_2^2$.
2. $\mathrm{P}(|\ \|\hat{\boldsymbol{x}}\|_2^2 - \|\boldsymbol{x}\|_2^2\ | \geq \epsilon \|\boldsymbol{x}\|_2^2) < 2\mathrm{e}^{-(\epsilon^2 - \epsilon^3)\frac{k}{4}}$.

The random matrix $\boldsymbol{R}$ defines a k-dimensional random subspace independent of the dataset X. The RP reduces the dimensionality without any consideration of the data distribution. This is a major difference of the RP from other feature selection and reduction techniques that involve mining important data attributes.

The generation of a dense random matrix requires a computational cost and memory storage of $\mathcal{O}(kd)$. Furthermore, the projection of N points requires $\mathcal{O}(kdN)$ operations. For practical computation, we use an efficient version of the RP [143] with operation and memory storage of $\mathcal{O}(d \log d)$.

3.3.3 Two-Dimensional Random Projection

For a set of two-dimensional arrays $\{\boldsymbol{X}_i\}_{i=1}^N$ such that $\boldsymbol{X}_i \in \mathbb{R}^{m \times n}$ and $\mathrm{E}(\boldsymbol{X}_i) = 0$, setting $\boldsymbol{R}_\mathrm{L} \in \mathbb{R}^{k_1 \times m}$ and $\boldsymbol{R}_\mathrm{R} \in \mathbb{R}^{k_2 \times n}$ to be RP matrices, we define the transform

$$\hat{\boldsymbol{X}}_i = \boldsymbol{R}_\mathrm{L} \boldsymbol{X}_i \boldsymbol{R}_R^\top. \tag{3.25}$$

For the set $\hat{X} = \{\hat{\boldsymbol{X}}_i\}_{i=1}^N$, we have the following theorem.

Theorem 3.1 *$\hat{\boldsymbol{X}}_i \in \hat{X}$ and $\boldsymbol{X}_i \in X$ satisfy the Johnson-Lindenstrauss property.*
(Proof) We set vec *to be an operator that vectorises a matrix. From $\hat{\boldsymbol{X}}_i = \boldsymbol{R}_{\rm L}\boldsymbol{X}_i\boldsymbol{R}_{\rm R}$, we have the relation*

$$\mathrm{vec}\hat{\boldsymbol{X}}_i = (\boldsymbol{R}_{\rm L} \otimes \boldsymbol{R}_{\rm R})\mathrm{vec}\boldsymbol{X}_i, \tag{3.26}$$

where $\boldsymbol{R}_{\rm L} \otimes \boldsymbol{R}_{\rm R} = \boldsymbol{R} \in \mathbb{R}^{k\times d}$ is an RP matrix. Here, $k = k_1 \times k_2$ and $d = m \times n$. Therefore, for any $\epsilon > 0$ and set X of N images $\{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$, $\hat{\boldsymbol{X}}_i$ and $\hat{\boldsymbol{X}}_j$ satisfy the property

$$(1-\epsilon)\|\boldsymbol{X}_j - \boldsymbol{X}_i\|_{\rm F} \leq \|\hat{\boldsymbol{X}}_j - \hat{\boldsymbol{X}}_i\|_{\rm F} \leq (1+\epsilon)\|\boldsymbol{X}_j - \boldsymbol{X}_i\|_{\rm F}. \tag{3.27}$$

Here, setting $\|\boldsymbol{A}\|_{\rm F}$ to be the Frobenius norm of matrix $\boldsymbol{A}$, the relation

$$\|\boldsymbol{x}_i\|_2^2 = \|\mathrm{vec}\boldsymbol{X}_i\|_2^2 = \|\boldsymbol{X}_i\|_{\rm F}^2 \tag{3.28}$$

is satisfied for $\boldsymbol{x}_i = \mathrm{vec}\boldsymbol{X}_i$. Therefore, by replacing the Euclidean norm of $\mathrm{vec}\boldsymbol{X}_i$ with the Frobenius norm of $\boldsymbol{X}_i$, we have the statement of the theorem. (Q.E.D.)

By manipulating a two-dimensional array as a second-order tensor, we can reduce the dimension of the tensorial data to an arbitrary dimension. Then, the 2DRP preserves the topology of the tensor in the function space since the Frobenius norm of the tensor is preserved. Furthermore, from Theorem 1, the 2DRP preserves the row and column distributions of two-dimensional arrays.

3.4 Topology and Geometry in Pattern Recognition

Setting Φ to be the transformation for a collection of sampled data $D = \{\boldsymbol{f}_k\}_{k=1}^n \subset \mathbb{R}^d$, $n \ll d$, we define the dimension dim $(\Phi(D))$ of the transformed space $\Phi(D)$ spanned by $\{\Phi(\boldsymbol{f}_k)\}_{k=1}^n$. We introduce the following definitions.

Definition 3.1 *If* dim $(\Phi(D)) <$ dim (D)*, we call Φ a dimension-reduction operation.*

Definition 3.2 *If $\Phi(\alpha\boldsymbol{f} + \beta\boldsymbol{g}) = \alpha\Phi(\boldsymbol{f}) + \beta\Phi(\boldsymbol{g})$ for a pair of scalars α and β, Φ is linear.*

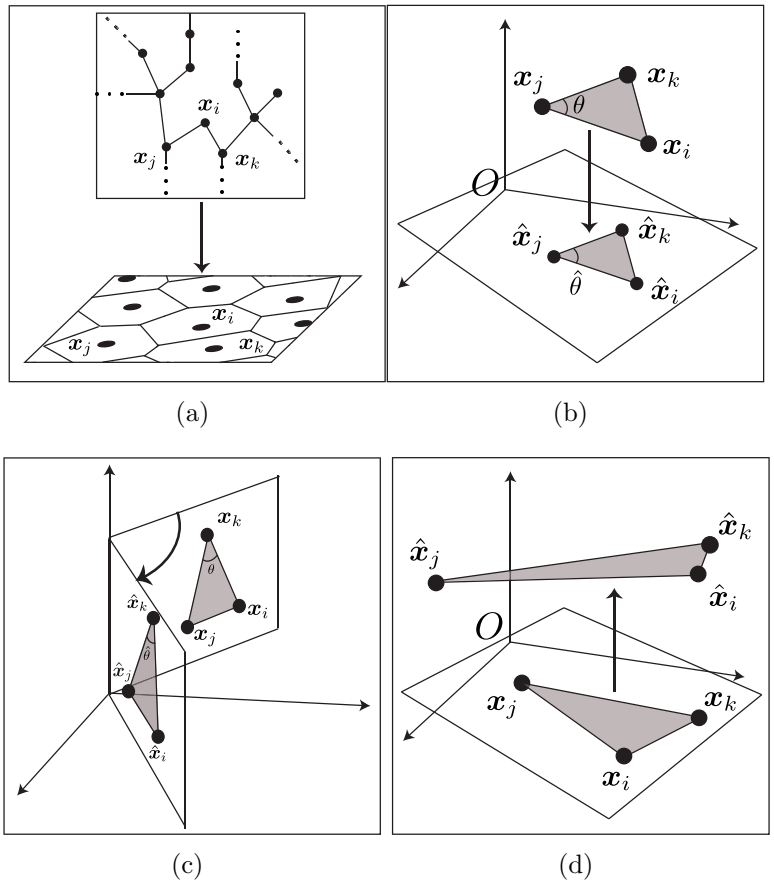


Figure 3.4: Requirements for a mapping in image pattern recognition. (a) Order condition. For classification including categorisation, dimension reduction should preserve the order structure in the original space. However, in many cases, data are embedded in a metric space and classified with respect to a metric. For example, in visual categorisation, histograms of images are embedded in a metric space. (b) Weak condition. The dimension reduction operation Φ approximately preserves distances and angles among data. (c) Strong condition. Φ preserves distances and angles among data. Only the rotation transform satisfies this condition. (d) Discriminative condition. Φ locally shrinks neighbourhoods and globally expands distances and angles among elements in the data space. This mapping increases the separation ratio between distributions of categories, although it is not a dimension reduction mapping.

Therefore, linear dimension reduction follows Definitions 3.1 and 3.2.

We set $d(\boldsymbol{f}, \boldsymbol{g})$ and $d_\Phi(\Phi(\boldsymbol{f}), \Phi(\boldsymbol{g}))$ as the metrics in $\mathbb{R}^d$ and $\Phi(D)$, respectively. Using d and d_Φ for the transform Φ, we define the topology-preserving property of Φ.

Definition 3.3 *Φ has a strong topology-preserving property if the condition $d(\boldsymbol{f}, \boldsymbol{g}) = d_\Phi(\Phi(\boldsymbol{f}), \Phi(\boldsymbol{g}))$ holds.*

Definition 3.4 *Φ has a weak topology-preserving property if the condition $|d(\boldsymbol{f}, \boldsymbol{g}) - d_\Phi(\Phi(\boldsymbol{f}), \Phi(\boldsymbol{g}))| < \epsilon$ holds for a small positive constant ϵ.*

Definition 3.5 *Setting T to be a positive threshold, for $\boldsymbol{f}$ and $\boldsymbol{g}$ such that $d(\boldsymbol{f}, \boldsymbol{g}) \geq T$, Φ is called an expansive mapping if $d_\Phi(\Phi(\boldsymbol{f}), \Phi(\boldsymbol{g})) \geq d(\boldsymbol{f}, \boldsymbol{g})$ holds.*

Definition 3.6 *Setting T to be a positive threshold, for $\boldsymbol{f}$ and $\boldsymbol{g}$ such that $d(\boldsymbol{f}, \boldsymbol{g}) \leq T$, Φ is called a nonexpansive mapping if $d_\Phi(\Phi(\boldsymbol{f}), \Phi(\boldsymbol{g})) \leq d(\boldsymbol{f}, \boldsymbol{g})$ holds.*

For both weak and strong topology-preserving transforms, we have the following propositions.

Proposition 3.2 *For strong topology-preserving operations in a vector space, we have $\angle(\boldsymbol{f}, \boldsymbol{g}) = \angle(\Phi(\boldsymbol{f}), \Phi(\boldsymbol{g}))$.*

(Proof) If Φ has strong topology-preserving properties, Φ is a rotation transform. The strong topology-preserving properties derive geometry-preserving properties since a rotation transform does not change the angle between data. (Q.E.D.)

Therefore, if the strong and weak conditions hold, the performance of distance- and angle-based classifications is invariant under dimension reduction. Otherwise, for the cases in Definitions 5 and 6, the results of classification before and after dimension reduction are different.

For a dimension-reduction operation used in pattern recognition, we define four requirements.

- Order condition: Dimension reduction is achieved by preserving orders in the neighbourhood of each element in a data space.
- Weak condition: Operation Φ approximately preserves distances and angles among data.
- Strong condition: Operation Φ preserves distances and angles among data.

- Discriminative condition: Operation Φ locally shrinks neighbourhoods as a nonexpansion mapping and globally expands distances and angles among elements in a data space as an expansive mapping. Φ increases the separation ratio between distributions of categories.

Figure 3.4 illustrates these requirements. The order condition is the fundamental requirement for dimension-reduction operations in pattern recognition. In spite of constructing the order structure, we usually embed data in a metric space. Therefore, the requirement of the order condition is replaced with the weak and strong conditions. Generally, in linear dimension-reduction operations, the strong and discriminative conditions do not hold. Among the linear dimension-reduction methods, the RP and the 2DRP satisfy the weak condition required in image pattern recognition. These two methods possess topology- and geometry-preserving properties. Other linear dimension-reduction methods do not satisfy the required weak condition to the best of our knowledge.

As a nonlinear dimension-reduction method, the kernel method maps data to a higher-dimensional space than the original data space by the mapping Φ. The kernel method satisfies the discriminative condition by introducing the operation

$$d_\Phi(\Phi(\boldsymbol{f}), \Phi(\boldsymbol{g})) \begin{cases} \ll d(\boldsymbol{f}, \boldsymbol{g}) & \text{if } d(\boldsymbol{f}, \boldsymbol{g}) < T_s, \\ \gg d(\boldsymbol{f}, \boldsymbol{g}) & \text{if } d(\boldsymbol{f}, \boldsymbol{g}) > T_l. \end{cases}, \tag{3.29}$$

which is a nonlinear mapping. For $T_s < T_l$, if $d(\boldsymbol{f}, \boldsymbol{g}) < T_s$ and $d(\boldsymbol{f}, \boldsymbol{g}) > T_l$, we have $d_\Phi(\Phi(\boldsymbol{f}), \Phi(\boldsymbol{g})) \ll d(\boldsymbol{f}, \boldsymbol{g})$ and $d_\Phi(\Phi(\boldsymbol{f}), \Phi(\boldsymbol{g})) \gg d(\boldsymbol{f}, \boldsymbol{g})$, respectively. Since this mapping Φ increases the separation ratio between distributions of data for two categories, we can establish a linear classification in the high-dimensional space.

For the practical computation of the kernel method, the kernel trick [149] gives the inner norm of two data in the projected high-dimensional space without computation of the norm or distance in the high-dimensional space. For example, for the kernel trick in ref. [149, 23], the polynomial function

$$k(\boldsymbol{f}, \boldsymbol{g}) = (\Phi(\boldsymbol{f}), \Phi(\boldsymbol{g})) = ((\boldsymbol{f}, \boldsymbol{g}) + 1)^p \tag{3.30}$$

and the Gaussian radial basis function

$$k(\boldsymbol{f}, \boldsymbol{g}) = (\Phi(\boldsymbol{f}), \Phi(\boldsymbol{g})) = \exp\left(-\frac{\|\boldsymbol{f} - \boldsymbol{g}\|_2^2}{2\sigma^2}\right) \tag{3.31}$$

give the inner norm of $\boldsymbol{f}$ and $\boldsymbol{g}$ in a high-dimensional space. These functions give a large inner norm for a pair with a larger distance than a threshold

T and vice versa. Using kernels, we perform the linear dimension-reduction method in a high-dimensional space and obtain a nonlinear map in the original space. However, note that Φ is not the dimension reduction operator since $\dim(\Phi(D)) \gg \dim(D)$. This indicates nonlinear dimension reduction based on the kernel trick is a compression method but not a dimension-reduction method.

These properties imply that the requirement for the dimension-reduction method is the weak condition.

3.5 Classification Methods

Defining the class subspace, we introduce four classification methods. In these classification methods, we construct discriminative class subspaces. For the construction of discriminative class subspaces, after preprocessing, topological information of the data distribution in the original space should be preserved because the following methods except for the 2DTSM adopt the angle between subspaces as a similarity. However, for the 2DTSM, preservation of the topology of an image is preferable.

3.5.1 Subspace Method

Setting $\mathbb{R}^d$ to be d-dimensional Euclidean space, we assume that the inner product $(\boldsymbol{f}, \boldsymbol{g})$ is defined in $\mathbb{R}^d$. Furthermore, we define the Schatten product $\langle \boldsymbol{f}, \boldsymbol{g} \rangle$, which is an operator from $\mathbb{R}^d$ to $\mathbb{R}^d$. Let $\boldsymbol{f} \in \mathbb{R}^d$ and $\boldsymbol{P}_k,\ i = 1, \ldots, N$, respectively be a pattern and an operator for the ith class, where the ith class is defined as

$$\mathcal{C}_i = \{\boldsymbol{f} \mid \boldsymbol{P}_i \boldsymbol{f} = \boldsymbol{f},\ \boldsymbol{P}_i^\top \boldsymbol{P}_i = \boldsymbol{I}\}. \tag{3.32}$$

Since patterns have perturbations, we define the ith class as

$$\mathcal{C}_i(\delta) = \{\boldsymbol{f} \mid \|\boldsymbol{P}_i \boldsymbol{f} - \boldsymbol{f}\|_2 \ll \delta,\ \boldsymbol{P}_i^\top \boldsymbol{P}_i = \boldsymbol{I}\}, \tag{3.33}$$

where δ is a small perturbation of the pattern and a small value. For input $\boldsymbol{g} \in \mathbb{R}^d$ and class $\mathcal{C}_i$, we respectively define the similarity and classification criteria as

$$\theta_i = \angle(C_i(\delta), \boldsymbol{g}),\ 0 < \theta_i < {}^{\exists}\theta_0 \rightarrow \boldsymbol{g} \in \mathcal{C}_i(\delta), \tag{3.34}$$

since we define the angle between input pattern $\boldsymbol{g}$ and the space of the pattern as

$$\theta_i = \cos^{-1} \frac{\|\boldsymbol{P}_i \boldsymbol{g}\|_2}{\|\boldsymbol{g}\|_2}. \tag{3.35}$$

The angle between the input pattern and the pattern space represents their similarity.

For input $\boldsymbol{g} \in \mathbb{R}^d$, we construct

$$\mathcal{C}_{\boldsymbol{g}} = \{\boldsymbol{g} \,|\, \boldsymbol{Q}\boldsymbol{g} = \boldsymbol{g},\, \boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{I}\}, \tag{3.36}$$

$$\mathcal{C}_{\boldsymbol{g}}(\delta) = \{\boldsymbol{g} \,|\, \|\boldsymbol{Q}\boldsymbol{g} - \boldsymbol{g}\|_2 \ll \delta,\, \boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{I}\}. \tag{3.37}$$

Then, we define a generalisation of Eq. (3.34) as

$$\theta_i = \angle(\mathcal{C}_i(\delta), \mathcal{C}_{\boldsymbol{g}}(\delta)),\, \theta < \theta_i < {}^{\exists}\theta_0 \rightarrow \mathcal{C}_{\boldsymbol{g}}(\delta) \in \mathcal{C}_i(\delta), \tag{3.38}$$

where $\sharp|\mathcal{C}_{\boldsymbol{g}} \backslash C_k(\delta) \cap \mathcal{C}_{\boldsymbol{g}}(\delta)| \ll \delta$.

We construct an operator $\boldsymbol{P}_i$ for $\boldsymbol{f}_i \in \mathcal{C}_i$ such that

$$\boldsymbol{P} = \arg\min\ (\mathrm{E}\|\boldsymbol{f} - \boldsymbol{P}_i\boldsymbol{f}\|_2) \quad \text{w.r.t.} \quad \boldsymbol{P}_i^\top \boldsymbol{P}_i = I, \tag{3.39}$$

where $f \in \mathcal{C}_i$, I is the identity operator and E is the expectation over $\mathbb{R}^d$.

For practical calculation, we set $\{\boldsymbol{\varphi}_j\}_{j=1}^n$ to be the eigenfunction of $\boldsymbol{M} = \mathrm{E}(\langle \boldsymbol{f}, \boldsymbol{f} \rangle)$. We define the eigenfunction of $\boldsymbol{M}$ as $\|\boldsymbol{\varphi}_j\|_2 = 1$ for eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_j \geq \cdots \geq \lambda_n$ with $n \leq d$. Therefore, the operator $\boldsymbol{P}$ is defined as $\boldsymbol{P} = \sum_{j=1}^n \langle \boldsymbol{\varphi}_j, \boldsymbol{\varphi}_j \rangle$.

Figure 3.5(a) shows the basic idea of the SM. To identify whether the input data are in the subspace of the classes, we calculate the angle between the input data and the subspace of the classes. If g belongs to the space, the length of the orthogonal projection is close to 1. Figure 3.5(b) shows multiclass recognition using the SM.

When we adopt the PT for preprocessing in the SM, the discriminative performance of this method basically deteriorates because differences among each subspace become small. However, for a class subspace that consists of nonsimilar patterns, upon adopting the PT, the discriminative performance of the SM becomes high because the PT contracts the class subspace.

3.5.2 Mutual Subspace Method

Let $\boldsymbol{P}_i$ and $\boldsymbol{Q}$ be operators for $\mathcal{C}_i$ and $\mathcal{C}_{\boldsymbol{g}}$, respectively. If a pattern is expressed as an element of the linear subspace $\mathcal{C}_{\boldsymbol{g}} = \{\boldsymbol{f} | \boldsymbol{Q}\boldsymbol{f} = \boldsymbol{f}, \boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{I}\}$, we are required to compute the angle between $\mathcal{C}_{\boldsymbol{g}}$ and C_i as an extension of classical pattern recognition such that $\operatorname{rank} \boldsymbol{Q} = 1$ and $\dim \mathcal{C}_{\boldsymbol{g}} = 1$. Then, the angle between $\boldsymbol{P}_i$ and $\boldsymbol{Q}$ is computed as

$$\cos\theta_i = \max \mathrm{E}\left(\frac{\|\boldsymbol{Q}\boldsymbol{P}_i\boldsymbol{f}\|_2}{\|\boldsymbol{f}\|_2}\right) = \max \mathrm{E}\left(\frac{\|\boldsymbol{P}_i\boldsymbol{Q}\boldsymbol{f}\|_2}{\|\boldsymbol{f}\|_2}\right), \tag{3.40}$$

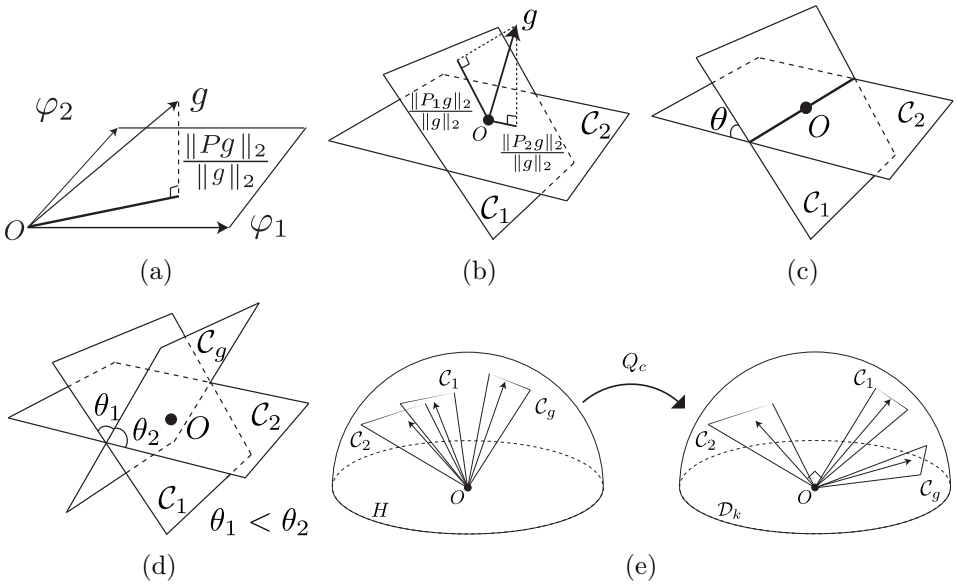


Figure 3.5: (a) Geometric properties of the subspace method (SM). Let $\boldsymbol{\varphi}_1$ and $\boldsymbol{\varphi}_2$ be the bases of a class pattern. For an input $\boldsymbol{g}$, similarity is defined as the orthogonal projection to the pattern space. (b) Multiclass recognition using the SM. Let $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ be operators for subspaces $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. The input $\boldsymbol{g}$ is labelled as being in the 1st class since subspace $\mathcal{C}_1$ has the longer projection length of $\boldsymbol{g}$. (c) Angle between two linear subspaces $\mathcal{C}_1$ and $\mathcal{C}_2$. The minimal angle between the two subspaces is 0. However, in the mutual subspace method (MSM), we adopt the angle θ to indicate the similarity between two subspaces. (d) Multiclass recognition using the MSM. For an input subspace $\mathcal{C}_{\boldsymbol{g}}$, let θ_1 and θ_2 be its angles relative to $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. The input subspace $\mathcal{C}_g$ is labelled as being in the 1st class since $\theta_1 < \theta_2$. (e) Projection onto constraint subspace. The triangles represent the subspace of categories $\mathcal{C}_1$ and $\mathcal{C}_2$ and the subspace of queries $\mathcal{C}_{\boldsymbol{g}}$. The left figure shows three subspaces in the pattern space $\mathbb{R}^d$. The right figure shows the subspaces in the constraint subspace $\mathcal{D}_k$. In the constraint subspace, the relation between $\mathcal{C}_1$ and $\mathcal{C}_2$ ideally becomes orthogonal since we omit the common subspace between subspaces $\mathcal{C}_1$ and $\mathcal{C}_2$.

where f satisfies $\|\boldsymbol{f}\|_2 \neq 0$. Figure 3.5(c) shows the angle between the two subspaces.

For the calculation, the following Theorem is applied [20].

Lemma 3.6 *The angle between $\mathcal{C}_i$ and $\mathcal{C}_g$ is calculated as the maximum of the eigenvalues of $\boldsymbol{P}_i\boldsymbol{Q}\boldsymbol{P}_i$ and $\boldsymbol{Q}\boldsymbol{P}_i\boldsymbol{Q}$.*

However, computations of the eigenvalues of $\boldsymbol{P}_i\boldsymbol{Q}\boldsymbol{P}_i$ and $\boldsymbol{Q}\boldsymbol{P}_i\boldsymbol{Q}$ are not effective. Since the dimension of the class subspace is smaller than that of the feature space, the ranks of $\boldsymbol{P}_i\boldsymbol{Q}\boldsymbol{P}_i$ and $\boldsymbol{Q}\boldsymbol{P}_i\boldsymbol{Q}$ are smaller than the dimension of the feature space. Therefore, we can effectively compute a problem equivalent to the eigenvalue decomposition of $\boldsymbol{P}_i\boldsymbol{Q}\boldsymbol{P}_i$ and $\boldsymbol{Q}\boldsymbol{P}_i\boldsymbol{Q}$. For operators $\boldsymbol{P}_i = \Sigma_{j=1}^{n} < \boldsymbol{\phi}_j\boldsymbol{\phi}_j >$ and $\boldsymbol{Q} = \Sigma_{j=1}^{m} < \boldsymbol{\psi}_j\boldsymbol{\psi}_j >$, we have

$$\boldsymbol{X} = x_{ij}, \quad x_{ij} = \sum_{l=1}^{m} (\boldsymbol{\psi}_i, \boldsymbol{\phi}_l)(\boldsymbol{\phi}_l, \boldsymbol{\psi}_j). \tag{3.41}$$

Solving the eigenvalue problem of $\boldsymbol{X}$ in Eq. (3.41), we obtain the angle between the two subspaces [62]. Figure 3.5(d) shows multiclass recognition using the MSM.

3.5.3 Constraint Mutual Subspace Method

We next define a common subspace. For $\boldsymbol{f} \neq \boldsymbol{g}$, in a common subspace $\boldsymbol{A}\boldsymbol{P}_C\boldsymbol{f} = \boldsymbol{P}_C\boldsymbol{g}$ and $\|\boldsymbol{A}\boldsymbol{P}_C\boldsymbol{f}\|_2 = \|\boldsymbol{P}_C\boldsymbol{g}\|_2$ are satisfied, where $\boldsymbol{A}$ and $\boldsymbol{P}_C$ are an appropriate equi-affine operation and orthogonal projection, respectively. All patterns in a common subspace are written in terms of the equi-affine transform. For the projections $\{\boldsymbol{P}_i\}_{i=1}^{N}$ to the classes $\{\mathcal{C}_i\}_{i=1}^{N}$, we have the operator for the common subspace

$$\boldsymbol{P}_C = \prod_{i=1}^{N} \boldsymbol{P}_i. \tag{3.42}$$

Therefore, we define the constraint subspace as the operator $\boldsymbol{Q}_C = \boldsymbol{I} - \boldsymbol{P}_C$, where I is the identity operator. Using the operator $\boldsymbol{Q}_C$, we can calculate the angle in the constraint subspace by Eq. (3.40). The orthogonal projection for the constraint subspace is a nonexpansive mapping.

In the constraint subspace, the angle $\theta_{C,i}$ between the projected reference subspace $\mathcal{C}_{C,i}, i = 1, \ldots, N$, and the projected input subspace $\mathcal{C}_{C,\boldsymbol{g}}$ is defined as

$$\cos\theta_{C,i} = \max \mathrm{E}\left(\frac{\|\boldsymbol{Q}_c\boldsymbol{Q}\boldsymbol{Q}_c\boldsymbol{P}_i\boldsymbol{f}\|_2}{\|\boldsymbol{f}\|_2}\right) \tag{3.43}$$

$$= \max \mathrm{E}\left(\frac{\|\boldsymbol{Q}_c\boldsymbol{P}_i\boldsymbol{Q}_c\boldsymbol{Q}\boldsymbol{f}\|_2}{\|\boldsymbol{f}\|_2}\right), \tag{3.44}$$

where $\boldsymbol{f}$ satisfies $\|\boldsymbol{f}\|_2 \neq 0$.

The subspace $\mathcal{D}_k$ is defined as the constraint subspace for the CMSM [57]. Figure 3.5(e) shows a scheme of the CMSM. Projecting $\mathcal{D}_k$, the class subspaces become orthogonal to each other. To construct the operator $\boldsymbol{Q}_c$ for $\mathcal{D}_k$, setting $\{\boldsymbol{\psi}_j\}_{j=1}^{N_C}$ to be the eigenfunction of $\boldsymbol{G} = \sum_{i=1}^{N} \boldsymbol{P}_i$, we define the eigenfunction of $\boldsymbol{G}$ as

$$\boldsymbol{G}\boldsymbol{\psi}_j = \lambda_j \boldsymbol{\psi}_j, \quad \|\boldsymbol{\psi}_j\|_2 = 1, \tag{3.45}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_j \geq \cdots \geq \lambda_{N_c}$ and $N_c = n \times N$. Using $\{\boldsymbol{\psi}_j\}_{j=1}^{N}$, the operator $\boldsymbol{Q}_c$ is defined as

$$\boldsymbol{Q}_C = \sum_{j=1}^{k} \langle \boldsymbol{\psi}_{N_c-(j-1)}, \boldsymbol{\psi}_{N_c-(j-1)} \rangle, \tag{3.46}$$

where $k < N_C$ and $k \leq d$. The dimension k of the difference subspace is selected experimentally.

If the dimension of the common space is unity and the basis of this space corresponds to the first eigenfunction, which is associated with the largest eigenvalue of the covariance of the pattern space, the operation is called the constant normalisation of patterns.

According to lemma 1, if an input pattern has high similarity to a class in the MSM, the input pattern has higher similarity to the class in the CMSM than to that in the MSM. However, the CMSM does not guarantee the preservation of dissimilarity according to [57]. The projection onto the constraint subspace is a nonexpansive mapping, therefore the angle between the two subspaces becomes small.

3.5.4 Tensor Subspace Methods

Two-Dimensional Tensor Subspace Method

As an extension of the subspace method for vector data, we introduced a new linear tensor subspace method for a matrix called the 2DTSM [81]. For a matrix $\boldsymbol{X}$, setting $\boldsymbol{P}_{\mathrm{L}}$ and $\boldsymbol{P}_{\mathrm{R}}$ to be orthogonal projections, we call the operation

$$\boldsymbol{Y} = \boldsymbol{P}_{\mathrm{L}}^{\top} \boldsymbol{X} \boldsymbol{P}_{\mathrm{R}} \tag{3.47}$$

the orthogonal projection of $\boldsymbol{X}$ to $\boldsymbol{Y}$. Therefore, using this expression for a collection of matrices $\{\boldsymbol{X}_i\}_{i=1}^{N}$, such that $\boldsymbol{X}_i \in \mathbb{R}^{m \times n}$ and $\mathrm{E}(\boldsymbol{X}_i) = 0$, the solutions of

$$\begin{aligned} (\boldsymbol{P}_{\mathrm{L}}, \boldsymbol{P}_R) = \arg\max \mathrm{E}\left(\frac{\|\boldsymbol{P}_{\mathrm{L}}^{\top} \boldsymbol{X}_i \boldsymbol{P}_{\mathrm{R}}\|_{\mathrm{F}}}{\|\boldsymbol{X}_i\|_{\mathrm{F}}} \right) \\ w.r.t.\ \boldsymbol{P}_{\mathrm{L}}^{\top} \boldsymbol{P}_{\mathrm{L}} = \boldsymbol{I},\ \boldsymbol{P}_{\mathrm{R}}^{\top} \boldsymbol{P}_{\mathrm{R}} = \boldsymbol{I} \end{aligned} \tag{3.48}$$

define a bilinear subspace that approximates $\{\boldsymbol{X}_i\}_{i=1}^N$. Here, the norm $\|\boldsymbol{X}\|_{\mathrm{F}}$ for matrix $\boldsymbol{X}$ represents the Frobenius norm. Therefore, using the solutions of Eq. (3.48), for $\mathcal{C}_i(\delta) = \{\boldsymbol{X} \mid \|\boldsymbol{P}_{\mathrm{L},i}^\top \boldsymbol{X} \boldsymbol{P}_{\mathrm{R},i} - \boldsymbol{X}\|_{\mathrm{F}} \ll \delta\}$, $i = 1, 2, \ldots, N_{\mathcal{C}}$, if an input data array $\boldsymbol{G}$ satisfies the condition

$$\arg\left(\max_i \frac{\|\boldsymbol{P}_{\mathrm{L},i}^\top \boldsymbol{G} \boldsymbol{P}_{\mathrm{R},i}\|_{\mathrm{F}}}{\|\boldsymbol{G}\|_{\mathrm{F}}}\right) = \{\boldsymbol{P}_{\mathrm{L},k}, \boldsymbol{P}_{\mathrm{R},k}\}, \tag{3.49}$$

we conclude that $\boldsymbol{G} \in \mathcal{C}_k(\delta)$.

In practical computation to find the projections P_{L} and P_{R} in Eq. (3.48), we adopt the MEV [131]. This is a projection considering the distributions of column and row vectors of sampled images.

3.6 Experiments

To evaluate the performance of the dimension-reduction methods for image pattern recognition, we compute the recognition rate using eight image datasets: cropped versions of the extended YaleB dataset [60], ORL face dataset [146], ETH80 dataset [103], NEC animal dataset [124], MNIST dataset [101], ETL9G character dataset [142] and CALTECH101 dataset [53] and classification and detection image sets of the PASCAL Visual Object Classes Challenge 2012 (VOC2012) development kit [52]. The YaleB and ORL datasets are for face recognition. The MNIST dataset is for handwritten digit recognition. The ETL9G dataset is for handwritten Chinese character recognition. The ETH80 and NEC animal datasets are for object recognition. The VOC2012 and CALTECH101 datasets are for both visual categorisation and object detection. Tables 3.2 and 3.3 list the details of the eight databases. Figure 3.6 shows examples of images from each database.

The highest recognition rates of the YaleB, ORL, ETH80, MNIST, ETL9G, CALTECH101 and VOC2012 are 97.7% [180], 98.9% [124], 91.7% [87], 99.5% [107], 99.3% [107], 91.4% [71] and 82.2% [51], respectively, to the best of our knowledge. These results were obtained using different preprocessing methods, training sets and classifiers from those in our validation. We could not find the highest recognition rate of the NEC animal since this dataset was originally constructed as a training set for deep learning. Note that the purpose of our evaluation is to observe the effects of dimension-reduction methods for image pattern recognition, not to obtain a higher recognition accuracy than those of the existing methods. For the evaluation, we adopt six dimension-reduction methods: the downsampling (DS), the pyramid transform (PT), the two-dimensional discrete cosine transform (2DDCT), the random projection (RP), the two-dimensional random projection (2DRP) and

Table 3.2: Details of each database. ♯class and ♯data/class represent the number of classes and the number of data in each class, respectively. The image size is the original size of the images in each dataset. The vectorised size is the size of the vectorised images. The reduced dimension is the dimension of the images after vector-representation-based dimension reduction. The reduced image size is the size of the images after image-representation-based dimension reduction. In the CALTECH101 and VOC2012, each image has a different resolution and aspect ratio. For evaluation, we downsampled images in the CALTECH101 and VOC2012 to 92×80 pixels and 111×142 pixels, respectively.

	♯class	♯data /class	image size [pixel]	vectorised size	reduced dimension	reduced image size [pixel]
YaleB	38	64	192×168	32,256	1024	32×32
ORL	40	10	112×92	10,304	1024	32×32
ETH80	30	41	128×128	16,384	1024	32×32
NEC	60	72	480×580	278,400	1024	32×32
MNIST	10	7,000	28×28	784	225	15×15
ETL9G	152	200	127×128	16,256	1024	32×32
CALTECH101	100	40-800	92×80	7,360	1024	32×32
VOC2012	20	300-4,100	111×142	15,762	1024	32×32

the metric MDS with Sammon's mapping (MDS). We adopt the metric MDS for the comparison of linear and nonlinear dimension-reduction methods [3]. For the MDS, we omit the MNIST, ETL9G, CALTECH101 and VOC2012 datasets since they contain too many images for practical computation using the MDS. The RP and MDS are applied to images after their vectorisation. The other dimension-reduction methods (DS, PT, 2DRP and 2DDCT) are applied before the vectorisation of images.

Before the computation of recognition rates, we evaluate the mean energy loss, the mean relative error of the distance between images and the cumulative contribution ratio of the six dimension-reduction methods. For a feature vector $\boldsymbol{x}$ and dimension-reduced feature vector $\hat{\boldsymbol{x}}$, using the Euclidean norm $\|\cdot\|_2$, we define the energy loss as $\mathrm{E}\,(1 - \|\hat{\boldsymbol{x}}\|_2^2/\|\boldsymbol{x}\|_2^2)$. For a two-dimensional array $\boldsymbol{X}$ and a dimension-reduced 2D array $\hat{\boldsymbol{X}}$, we define the energy loss as $\mathrm{E}\,(1 - \|\hat{\boldsymbol{X}}\|_{\mathrm{F}}^2/\|\boldsymbol{X}\|_{\mathrm{F}}^2)$. Here, $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm. For feature vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ and dimension-reduced feature vectors $\hat{\boldsymbol{x}}_i$ and $\hat{\boldsymbol{x}}_j$, we define the

[3]The MDS embeds data into a low-dimensional space. This embedding is a nonlinear dimension-reduction method, while the kernel method uses a linear dimension-reduction method in a high-dimensional space.

Table 3.3: Conditions of images in each dataset. This table summarises whether an image in each dataset includes cropping of the region of interest, centring of the target in the image, changes in illumination, changes in camera position and the same background. Furthermore, the last term "same object" shows whether images are taken of the same object. ◯ indicates the satisfaction of a condition. △ indicates that a condition is partly satisfied in a dataset. × indicates the nonsatisfaction of a condition.

	cropping	centring	illumination changes	camera position	same background	same object
YaleB	◯	◯	◯	×	◯	◯
ORL	◯	◯	×	◯	◯	△
ETH80	◯	◯	×	◯	◯	◯
NEC	×	×	×	◯	◯	◯
MNIST	◯	◯	×	×	◯	×
ETL9G	◯	×	×	×	◯	×
CALTECH101	△	△	◯	◯	×	×
VOC2012	×	×	◯	◯	×	×

relative error of the distance as $\mathrm{E}\left((|\|\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_j\|_2 - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2)/\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2\right)$. Moreover, for two-dimensional arrays $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ and dimension-reduced two-dimensional arrays $\hat{\boldsymbol{X}}_i$ and $\hat{\boldsymbol{X}}_j$, we define the relative error of a two-dimensional array as $\mathrm{E}\left((|\|\hat{\boldsymbol{X}}_i - \hat{\boldsymbol{X}}_j\|_{\mathrm{F}} - \|\boldsymbol{X}_i - \boldsymbol{X}_j\|_{\mathrm{F}}|)/\|\boldsymbol{X}_i - \boldsymbol{X}_j\|_{\mathrm{F}}\right)$. Using the eight datasets and the six dimension-reduction methods, we compute the energy loss and relative error of 1000 randomly selected images. For a set of eigenvalues $\{\lambda_i\}_{i=1}^{N_\lambda}$ obtained by the PCA, the cumulative contribution ratio from first to the lth eigenvalue is defined as $\sum_{i=1}^{l}\lambda_i \sum_{i=1}^{N_\lambda}\lambda_i$. For the eight datasets and six dimension-reduction methods, we compute the cumulative contribution ratio using the original and dimension-reduced images of a class in each dataset. If the difference between the cumulative contribution ratios before and after dimension reduction is sufficiently small, a dimension-reduction method preserves the geometry among the data since the inner norm among the data is preserved.

For the evaluation of two-dimensional image decompsitions, we compare the CCRs of the marginal eigenvector (MEV) and generalised principal component analysis (GPCA). As the GPCA, we adopt the full projection (FP) and full projection truncation (FPT) described in Algorithm 1.1. For the sets of eigenvalues $\{\lambda_i^r\}_{i=1}^{N_\lambda^r}$ and $\{\lambda_i^c\}_{i=1}^{N_\lambda^c}$ for modes 1 and 2 obtained by the 2DSVD, the CCRs for modes 1 and 2 from the first eigenvalue to the lth eigenvalue are defined as $\sum_{i=1}^{l}\lambda_i^r/\sum_{i=1}^{N^r}\lambda_i^r$ and $\sum_{i=1}^{l}\lambda_i^c/\sum_{i=1}^{N^c}\lambda_i^c$, respectively. More-

Table 3.4: Summary of evaluation of dimension-reduction methods. For the evaluation of the energy loss and relative error, ◯, △ and × represent preservation with a small compression ratio, preservation and no preservation, respectively. For the evaluation of the cumulative contribution ratio, we give some remarks. In this table, DS, PT, RP, 2DRP, 2DDCT and MDS are abbreviations for the downsampling, pyramid transform, random projection, two-dimensional random projection, two-dimensional discrete cosine transform and multidimensional scaling, respectively.

dataset	classifier	energy loss	distance between images	cumulative contribution ratio
YaleB	DS	×	×	The PT gives a larger cumulative contribution ratio than the DS, RP, 2DRP, 2DDCT and MDS for a small compression ratio. The MDS gives the smallest cumulative contribution ratio.
	PT	×	×	
	RP	◯	◯	
	2DRP	△	△	
	2DDCT	◯	◯	
	MDS	×	◯	
ORL	DS	×	×	The PT gives a larger cumulative contribution ratio than the DS, RP, 2DRP, 2DDCT and MDS for a small compression ratio. The MDS gives the smallest cumulative contribution ratio.
	PT	×	×	
	RP	◯	◯	
	2DRP	×	△	
	2DDCT	◯	△	
	MDS	×	◯	
ETH80	DS	×	×	The PT and 2DRP give larger cumulative contribution ratios than the DS, RP, 2DDCT and MDS for a small compression ratio. The cumulative contribution ratio of the PT is the largest. The MDS gives the smallest cumulative contribution ratio.
	PT	×	×	
	RP	◯	◯	
	2DRP	△	△	
	2DDCT	◯	△	
	MDS	×	◯	
NEC	DS	×	×	The 2DRP gives a larger cumulative contribution ratio than the DS, PT, RP, 2DDCT and MDS for a small compression ratio.
	PT	×	×	
	RP	◯	◯	
	2DRP	△	◯	
	2DDCT	◯	△	
	MDS	×	◯	
MNIST	DS	×	×	The PT gives a larger cumulative contribution ratio than the DS, RP, 2DRP and 2DDCT for a small compression ratio.
	PT	×	×	
	RP	◯	◯	
	2DRP	△	△	
	2DDCT	×	×	
ETL9G	DS	×	×	The PT and 2DRP give larger cumulative contributions than the DS, RP and 2DDCT for a small compression ratio.
	PT	×	×	
	RP	◯	◯	
	2DRP	×	△	
	2DDCT	×	×	
CALTECH101	DS	×	×	The PT gives a larger cumulative contribution ratio than DS, the RP, 2DRP and 2DDCT for a small compression ratio.
	PT	×	×	
	RP	◯	◯	
	2DRP	×	△	
	2DDCT	◯	△	
VOC2012	DS	×	×	The PT gives a larger cumulative contribution ratio than the DS, RP, 2DRP and 2DDCT for a small compression ratio.
	PT	×	×	
	RP	◯	◯	
	2DRP	×	△	
	2DDCT	◯	△	

Table 3.5: Dimensions of the class subspaces used in the classification. ♯query represents the number of queries for each category. ♯basis represents the number of bases used for each category in recognition. The dimension of the class subspace represents the dimension of the constraint subspace.

dataset	classifier	♯query	♯basis	dimension of class subspace
YaleB	SM	1	1-32	-
	MSM	3, 5, 7	3, 5, 7	-
	CMSM	3, 5, 7	3, 5, 7	938, 950, 960, . . . , 1000, 1024
	2DTSM	1	1×1-32×32	-
ORL	SM	1	1-5	-
	MSM	3, 5	3, 5	-
	CMSM	3, 5	3, 5	938, 950, 960, . . . , 1000, 1024
	2DTSM	1	1×1-32×32	-
ETH80	SM	1	1-21	-
	MSM	3, 5, 7	3, 5, 7	-
	CMSM	3, 5, 7	3, 5, 7	938, 950, 960, . . . , 1000, 1024
	2DTSM	1	1×1-32×32	-
NEC	SM	1	1-36	-
	MSM	3, 5, 7	3, 5, 7	-
	CMSM	3, 5, 7	3, 5, 7	938, 950, 960, . . . , 1000, 1024
	2DTSM	1	1×1-32×32	-
MNIST	SM	1	1-225	-
	MSM	3, 5, 7	3, 5, 7	-
	CMSM	3, 5, 7	3, 5, 7	10, 20, . . . , 220, 225
	2DTSM	1	1×1-15×15	-
ETL9G	SM	1	1-1024	-
	MSM	3, 5, 7	3, 5, 7	-
	CMSM	3, 5, 7	3, 5, 7	938, 950, 960, . . . , 1000, 1024
	2DTSM	1	1×1-32×32	-
CALTECH101	SM	1	1-25	-
	MSM	3, 5, 7	3, 5, 7	-
	CMSM	3, 5, 7	3, 5, 7	938, 950, 960, . . . , 1000, 1024
	2DTSM	1	1×1-32×32	-
VOC2012	SM	1	1-25	-
	MSM	3, 5, 7	3, 5, 7	-
	CMSM	3, 5, 7	3, 5, 7	938, 950, 960, . . . , 1000, 1024
	2DTSM	1	1×1-32×32	-

over, before the comparison among the MEV, FP and FPT, we observe the convergence of Algorithm 1.1 for the eight datasets.

Figures 3.7-3.9 show the mean energy loss, the mean relative error and cumulative contribution ratio for each combination of dimension-reduction method and dataset, respectively. In these figures, the horizontal axis represents the compression ratio (C.R.), defined as C.R. = (original dimension) / (compressed dimension), where the uncompressed dimension represents the size of the vectorised images in Table 1. Table 3.4 summarises the results in Figs. 3.7-3.9. Figure 3.10 shows the sum of energies Ψ_k of all the projected images in each step of the iterations in Algorithm 1.1. Figure 3.11 shows the CCRs of decompositions by the FP, FPT and MEV for modes 1 and 2.

Figure 3.7 shows that for all the datasets, the RP and 2DRP preserve the energy of an image even for a large compression ratio. The DS, PT, 2DDCT and MDS do not preserve the energy of an image for a large compression ratio. The energy loss for the PT is higher than that for the DS. As shown in Fig. 3.8, for all the datasets, the RP, 2DRP and MDS have smaller relative errors than the DS, PT and 2DDCT. The 2DDCT has a small relative error for the YaleB, ORL, ETH80, NEC, CALTECH101 and VOC2012. However, it has large relative errors for images with a large compression ratio because the images of characters have high energy in their minor principal vectors. As shown in Fig. 3.9, for all the datasets, the PT gives a higher cumulative contribution ratio than the DS, RP, 2DRP, 2DDCT and MDS. The MDS gives a lower cumulative contribution ratio than the other methods for the YaleB, ORL and ETH80. However, the cumulative contribution ratios for the DS, RP and no compression are coincident. This result shows the partial topology-preserving property of the DS and the topology-preserving property of the RP. The MDS preserves pairwise distances for only given images but does not preserve the geometry in the original feature space. Furthermore, the results in Fig. 3.10 show that the algorithms used to compute the FP and FPT output the solution after the first iteration for all eight datasets. This fast convergence implies that the projection matrices computed by the MEV are approximately equivalent to those computed by the GPCA. As shown in Fig. 3.11, the principal eigenvalues computed by the MEV and FP coincide with each other in both mode 1 and mode 2. The principal eigenvalues from the principal major to the 34th major computed by the FPT are larger than those computed by the MEV and FP. However, the graphs of the cumulative contribution ratios for the three methods, MVE, FP and FPT, are approximately the same.

We calculate the recognition rates using the original and dimension reduced images of the eight datasets. For dimension reduction, we use six methods. In MNIST and VOC2012, the images were divided into training

Table 3.6: Summary of results of recognition rates. In this table, SM, MSM, CMSM and 2DTSM are abbreviations for the subspace method, mutual subspace method, constraint mutual subspace method and two-dimensional tensor subspace method, respectively.

	classifier	remarks
YaleB	SM	Low-dimensional linear subspaces of classes exist and contribute to classification.
	MSM	Angles among class subspaces contribute to classification.
	CMSM	There is a common linear subspace for classes.
	2DTSM	Peaks of the recognition rate appear at dimensions of 0.3-1.2% of that of the feature space.
ORL	SM	Low-dimensional linear subspaces of classes exist and contribute to classification.
	MSM	Angles among class subspaces contribute to classification.
	CMSM	There is a common linear subspace for classes.
	2DTSM	Peaks of the recognition rate appear at dimensions of less than 1% of that of the feature space.
ETH80	SM	None of the class subspaces contribute to classification.
	MSM	Angles among class subspaces contribute to classification.
	CMSM	There is no common linear subspace for classes.
	2DTSM	Peaks of the recognition rate appear at dimensions of less than 1.2% of that of the feature space.
NEC	SM	Low-dimensional linear subspaces of classes exist and contribute to classification.
	MSM	Angles among class subspaces do not contribute to classification.
	CMSM	There is a partly common subspace. The common subspace contributes to classification.
	2DTSM	Peaks of the recognition rate appear at dimensions of less than 0.08% of that of the feature space.
MNIST	SM	Low-dimensional linear subspaces of classes exist and contribute to classification.
	MSM	Angles among class subspaces contribute to recognition.
	CMSM	There is no common linear subspace for classes.
	2DTSM	Peaks of the recognition rate appear at dimensions of less than 3% of that of the feature space.
ETL9G	SM	Low-dimensional linear subspaces of classes exist and contribute to classification.
	MSM	Angles among class subspaces contribute to recognition.
	CMSM	There is no common linear subspace for classes.
	2DTSM	Peaks of the recognition rate appear at dimensions of less than 1% of that of the feature space.
CALTECH101	SM	None of the class subspaces contribute to classification.
	MSM	Angles among class subspaces contribute to recognition.
	CMSM	There is no common linear subspace for classes.
	2DTSM	There are no peaks of the recognition rate. There is no tensorial subspace for classification.
VOC2012	SM	None of the class subspaces contribute to classification.
	MSM	Angles among class subspaces do not contribute to recognition.
	CMSM	There is no common linear subspace for classes.
	2DTSM	There are no peaks of the recognition rate. There is no tensorial subspace for classification.

and test data by the distributor in advance. For the YaleB, ORL, ETH80, NEC, ETL9G and CALTECH101, we use the images labelled with even numbers as training data and the other images as test data. The recognition rate is defined as the successful label estimation ratio for 1000 label estimations. In each estimation, queries are randomly chosen from the test data. For recognition, we use the SM, MSM, CMSM and 2DTSM as classifiers. In the SM, MSM and CMSM, we use the vectors representing the image as a feature. In the 2DTSM, we use the matrices representing the image as a feature. For the 2DTSM, we additionally use the MEV, FP and FPT for the dimension reduction of images to compare the results of compression methods for two-dimensional images.

Figures 3.12-3.19 respvectively summarise the recognition rates of the YaleB, ORL, ETH80, NEC, MNIST, ETL9G, CALTECH101 and VOC2012 for the original and dimension-reduced images using the three classifiers. Figures 3.20 and 3.21 summarise the recognition rates of the SM and 2DTSM for the eight datasets, respectively. In Figs. 3.12-3.21, the horizontal axis represents the compression ratio, where the dimensions before and after compression are the numbers of reduced dimensions in Table 2 and bases in Table 5, respectively.

According to Figs. 3.12 and 3.13, when using five or more bases, the MSM accomplishes a recognition rate of 100% for the face recognition datasets. Furthermore, using the CMSM, we can accomplish 100% recognition with fewer than five bases. According to these results, the PT gives a smaller recognition rate than the DS, RP, 2DRP and 2DDCT. As shown in Figs. 3.14 and 3.15, none of the combinations of a dimension-reduction method and a classifier can accomplish 100% recognition. Even using the CMSM, we cannot obtain a larger recognition rate than that obtained by the MSM. In these cases, the PT gives a larger recognition rate than the DS, RP, 2DRP and 2DDCT. In Figs. 3.16 and 3.17, the MSM accomplishes 100% recognition. However, the CMSM does not make any contribution to the increase of recognition rate. In the case of the MNIST, the PT gives a smaller recognition rate than the DS, RP, 2DRP and 2DDCT. In contrast, for the ETL9G dataset, the PT gives a larger recognition rate than the DS, RP, 2DRP and 2DDCT. As shown in Figs. 3.18 and 3.19, the maximum recognition rates of both the CALTECH101 and VOC2012 are less than 60% and 15%, respectively. For these datasets, the CMSM does not make any contribution to the increase of recognition rate. Table 3.6 summarises the results of the computation of the recognition rates.

As shown in Fig. 3.21, for the 2DTSM, all dimension-reduction methods except the 2DRP give approximately the same results for each dataset. These results illustrate that the MEV, FP, FPT and 2DDCT give the same

recognition rate. The comparison of Figs. 3.20 and 3.21 shows that for all the datasets, the matrix representation gives lower recognition rates than the vector representation.

As shown in Fig. 3.9, for all datasets, the PT has the highest cumulative contribution in a low-dimensional linear subspace. Since the PT is a non-expansive mapping, the distances among the data become small. However, Fig. 3.7 illustrates that for the PT, a large amount of energy is lost in the images. Furthermore, Fig. 3.8 shows that the PT has large relative errors. In contrast, the RP and 2DRP preserve energies and distances. From the results in Figs. 3.12 and 3.13, we can observe that face images have a linear subspace enabling the recognition of face patterns. Moreover, face images have a common structure that prevents accurate recognition. For face recognition with an appropriate linear subspace, the PT is not an appropriate dimension-reduction method because it does not preserve the topology and geometry of the feature space. As shown in Figs. 3.14 and 3.15, object images do not have a linear subspace enabling recognition or a common structure. The PT gives a larger recognition rate than the DS, RP, 2DRP and 2DDCT for some cases and a smaller recognition rates for other case. Figures 3.16 and 3.17 show that characters have a linear subspace enabling recognition but do not have a common structure. For the MNIST, the PT gives a smaller recognition rate than the DS, RP, 2DRP and 2DDCT. For the ETL9G, in contrast, the PT gives a larger recognition rate than the DS, RP, 2DRP and 2DDCT. The results in Figs. 3.18 and 3.19 show that there is no linear subspace enabling recognition of visual categories. Therefore, we conclude that the global features of an image are not suitable for the recognition of visual categories. As shown in Figs. 3.10, 3.11 and 3.21, these properties of the principal components, compression and recognition for data computed by the three methods, MVE, FP and FPT, imply that, for the practical computation of 2DSVD, the MVE, which is a non-iterative method, is sufficient. Furthermore, Fig. 3.21 shows that the 2DDCT is an acceptable approximation for the 2DSVD since the recognition ratios by 2DDCT, MEV and GPCA are almost the same. From Figs. 3.20 and 3.21, we conclude that the vector representation gives higher recognition rates because the topology of the vector space makes a larger contribution to pattern recognition than that for the matrix representation. In Figs. 3.12-3.20, the recognition rates of the randomly projected images are approximately coincidnet to those of the original image with high probability. In Figs. 3.12-3.15, the MDS gives different curves for the recognition rates from those of the RP.

From these experiments, we clarified the following properties.

- The PT has a higher cumulative contribution ratio than the DS, RP,

2DRP and 2DDCT.

- The PT has a smaller and larger recognition rate than the RP and DS, respectively. The PT preserves the local geometry and topology of an image. However, it changes the distances and angles among the vectors used as data since it is a nonexpansive mapping.
- The MDS only preserves pairwise distances for given images and does not preserve the geometry of the original feature space.
- The 2DDCT is an acceptable approximation for the TPCA and 2DSVD.
- The DS, PT, 2DRP and 2DDCT have approximately the same recognition rate.
- The RP and MDS have different recognition rates, since the MDS does not preserve a geometry in a feature space.
- Vector-based recognition has a higher recognition rate than matrix-based recognition.
- For the practical selection of the method of dimension reduction, the RP works well compared with the DS, PT, 2DRP, 2DDCT and MDS if we have no *a priori* information on the input data since the RP preserves the topology and geomerty of the original feature space.

3.7 Summary

We mathematically analysed and experimentally evaluated the validity of dimension-reduction methods for image pattern recognition.

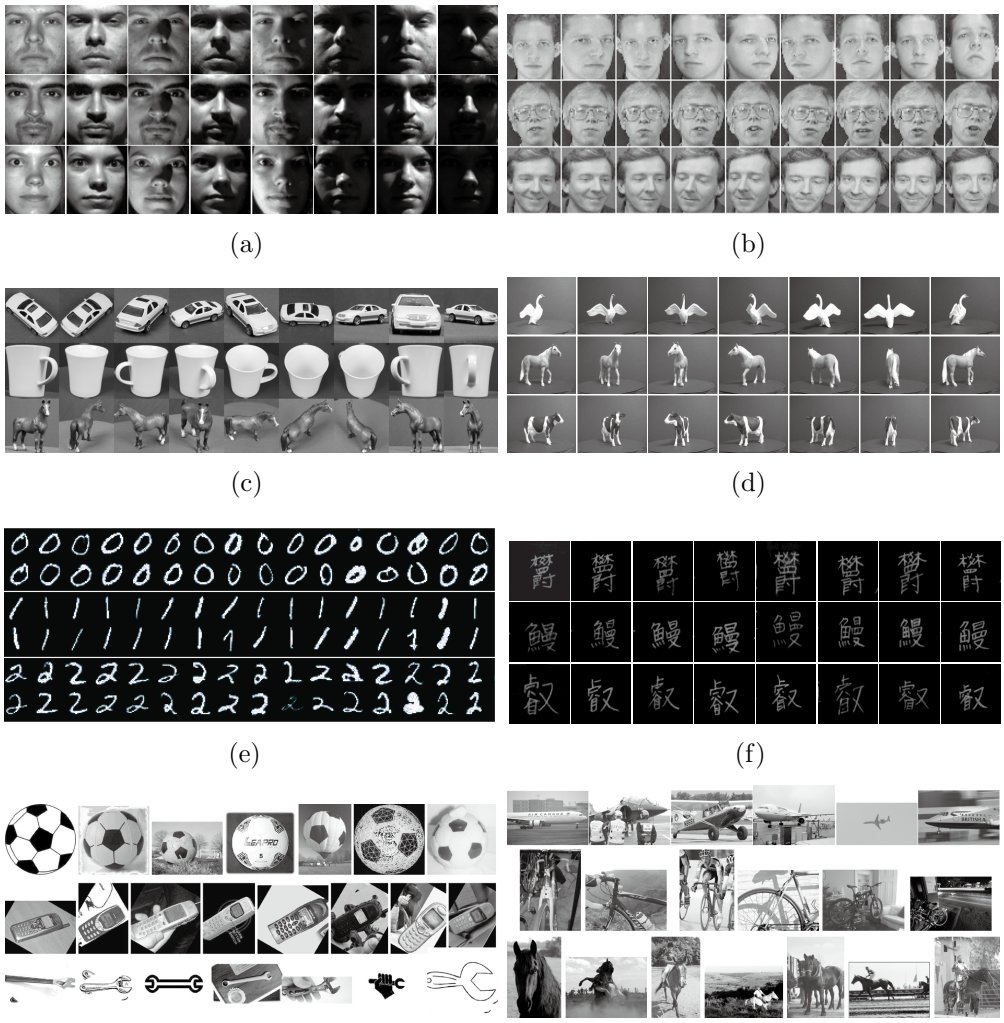
We defined four essential conditions for dimension reduction for image pattern recognition. By clarification of the nonexpansive mapping and topology-preserving mapping, we showed that only the topology-preserving mapping preserves distances and angles among data. The approximate preservation of distances and angles among data is the weak condition for dimension-reduction methods. We concluded that the weak condition is only satisfied by the random projection among the linear dimension-reduction methods. The pyramid transform and other methods do not satisfy any essential conditions.

By the evaluation of the dimension-reduction operation in experiments, we clarified the following properties. Firstly, for eight databases, the pyramid transform has a higher cumulative contribution ratio than the downsampling,

random projection, two-dimensional random projection, two-dimensional discrete cosine transform and multidimensional scaling. The multidimensional scaling has a lower cumulative contribution ratio than the downsampling, pyramid transform, random projection, two-dimensional random projection and two-dimensional discrete cosine transform. These comparisons imply that the geometry of an original feature space is not preserved by the pyramid transform and nonlinear multidimensional scaling. Secondly, using feature vectors for recognition, the pyramid transform has a different recognition rate from the random projection. These results show that the pyramid transform changes the distances and angles among feature vectors since it is a nonexpansive mapping. Thirdly, when using global features of an image, the downsampling, two-dimensional random projection, two-dimensional discrete cosine transform and nonlinear multidimensional scaling have approximately the same recognition rate. Finally, vector-based recognition has a higher recognition rate than matrix-based recognition. From this property, the classification should be computed in a vector space.

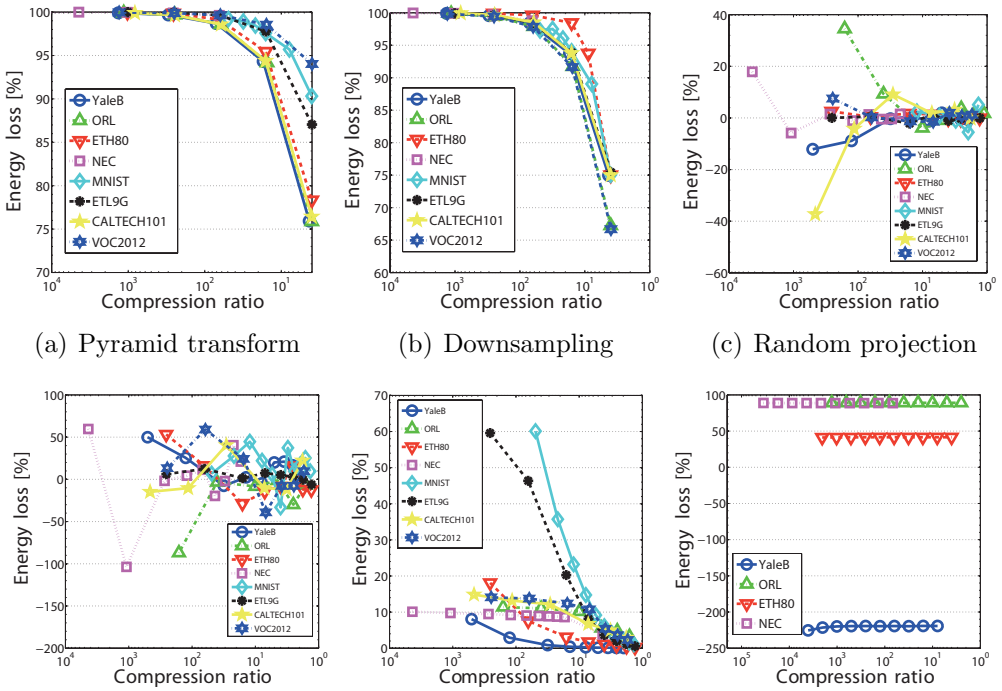
Nonexpansive mapping, that is, the pyramid transform, has a different recognition rate before and after dimension reduction. This property may lead to misunderstanding of the performance of classifiers. Therefore, for pattern recognition, we should use a dimension-reduction method that preserves the topology of the vector space. In contrast to the pyramid transform, the random projection preserves the topology and geometry of the vector space. This property implies that the random projection works well as a dimension-reduction method compared with other methods if we have no *a priori* information on the input data. Therefore, for images whose properties are not clear, we should adopt the random projection as the dimension-reduction method.

In this chapter, we surveyed linear problems and comparatively evaluated the performance of linear methods for pattern recognition. We clarified that a classifier can achieve a higher recognition rate if we use nonexpansive mapping for preprocessing. This property is a common property to both linear and nonlinear methods. Moreover, we deal with the pyramid transform as one of the well-known linear transforms, whereas a simple nonlinear transform is the normalisation.



(a) (b) (c) (d) (e) (f) (g) (h)

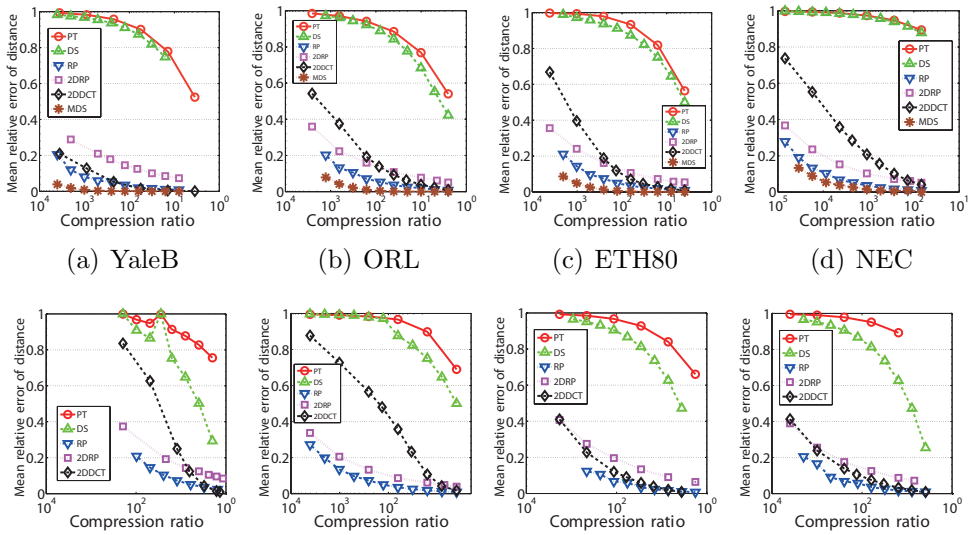
Figure 3.6: Examples of three categories in each dataset. (a) YaleB. (b) ORL. (c) ETH80. (d) NEC. (e) MNIST. (f) ETL9G. (g) CALTECH101. (h) VOC2012.



(a) Pyramid transform (b) Downsampling (c) Random projection

(d) 2D random projection (e) 2D discrete cosine transform (f) Multidimensional scaling

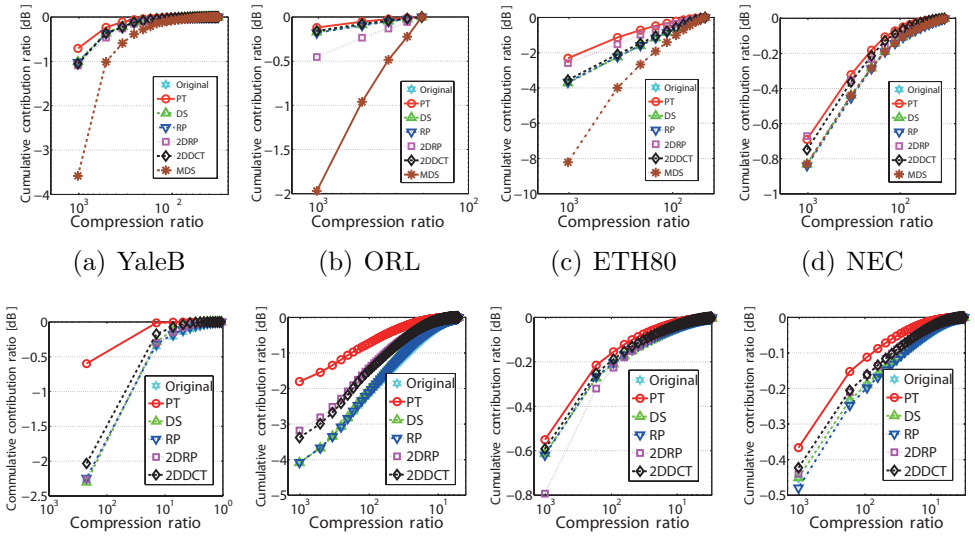
Figure 3.7: Energy loss of dimension-reduction methods. (a)-(f) show the energy loss for the pyramid transform, downsampling, random projection, two-dimensional random projection, two-dimensional discrete cosine transform and metric multidimensional scaling, respectively. In these figures, circles, upward triangles, downward triangles, squares, diamonds, asterisks, five-pointed stars and six-pointed stars represent the YaleB, ORL, ETH80, NEC, MNIST, ETL9G, CALTECH101 and VOC2012 datasets, respectively. The horizontal and vertical axes represent the compression ratio and energy loss, respectively.



(a) YaleB (b) ORL (c) ETH80 (d) NEC

(e) MNIST (f) ETL9G (g) CALTECH101 (h) VOC2012

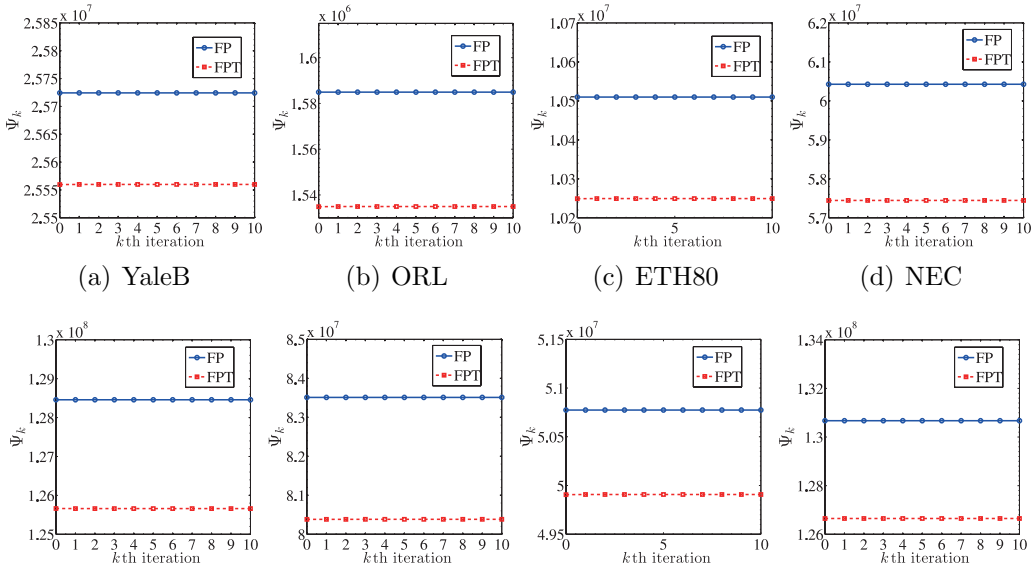
Figure 3.8: Mean relative error between distances in original space and dimension-reduced space. In the computation, we randomly select 1000 pairs from each dataset. In (a)-(h), circles, upward triangles, downward triangles, squares, diamonds and asterisks represent the pyramid transform, downsampling, random projection, two-dimensional random projection, two-dimensional discrete cosine transform and metric multidimensional scaling, respectively. The horizontal and vertical axes represent the compression ratio and relative error, respectively.



(a) YaleB (b) ORL (c) ETH80 (d) NEC

(e) MNIST (f) ETL9G (g) CALTECH101 (h) VOC2012

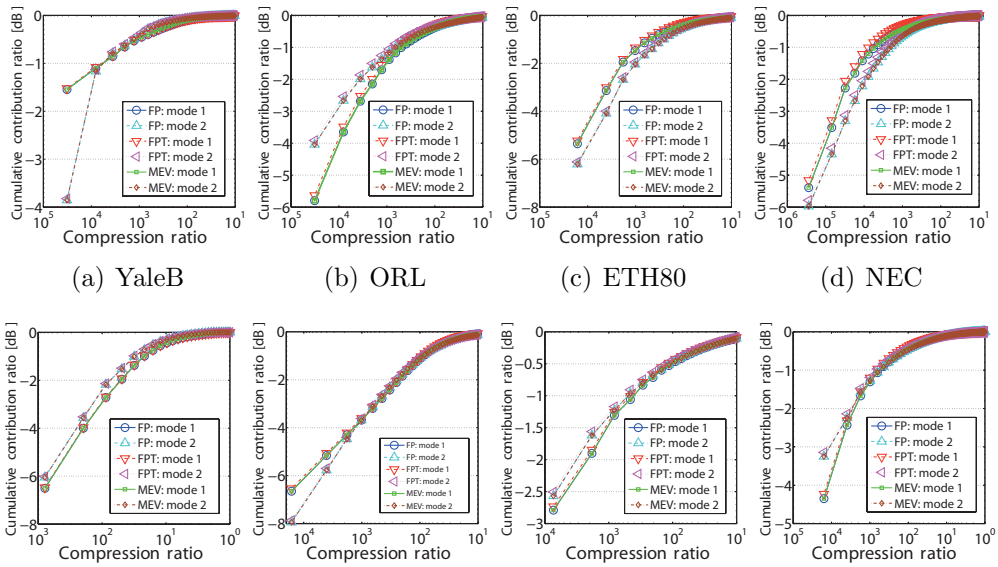
Figure 3.9: Cumulative contribution ratios. In (a)-(h), stars, circles, upward triangles, downward triangles, squares, diamonds and asterisks represent the cumulative contribution ratios of the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection, two-dimensional discrete cosine transform and metric multidimensional scaling, respectively. The horizontal and vertical axes represent the compression ratio and cumulative contribution ratio, respectively. The cumulative contribution ratio is displayed as a logarithm to base 10.



(a) YaleB (b) ORL (c) ETH80 (d) NEC

(e) MNIST (f) ETL9G (g) Caltech (h) VOC

Figure 3.10: Convergences of full projection method and full projection truncation method. In (a)-(h), circles and squares represent the sum of energies of all projected images in each step of the iteration for the full projection (FP) method and full projection truncation (FPT) method, respectively. Horizontal and vertical axes present the number of iterations and the sum of energies of all projected images, respectively.



(a) YaleB (b) ORL (c) ETH80 (d) NEC

(e) MNIST (f) ETL9G (g) Caltech (h) VOC

Figure 3.11: Cumulative contribution ratios. In (a)-(b), circles, downward triangles and squares represent the cumulative contribution ratio of the eigevalues of mode 1 for the full projection (FP) method, full projection truncation (FPT) method and marginal eigenvector (MEV) method, respectively. In (a)-(b), upward triangles, leftward triangles and diamonds represent the cumulative contribution ratio of the eigenvalues of mode 2 for the FP method, FPT method and MEV method, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and cumulative contribution ratio, respectively. The cumulative contribution ratio is displayed as a logarithm to base 10. The compression ratio is the ratio to the original size of the images in Table 3.2.

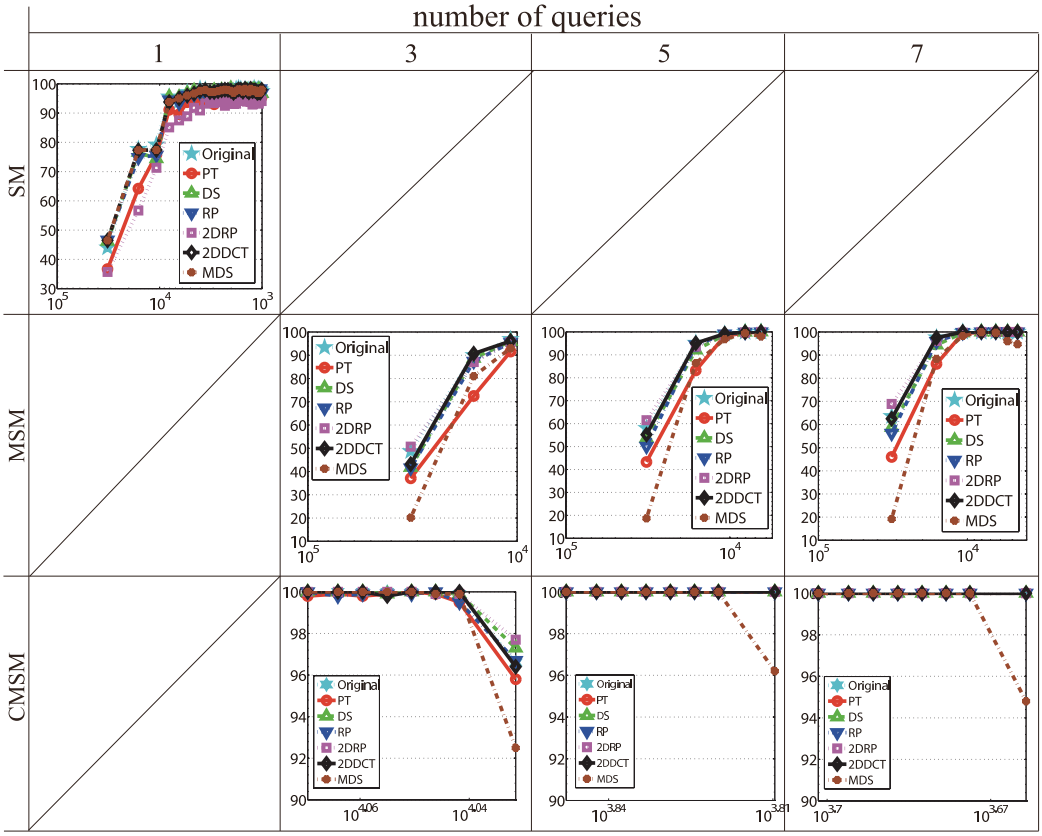


Figure 3.12: Recognition rates for each pair consisting of dimension-reduction method and classification method in the YaleB dataset. Stars, circles, upward triangles, downward triangles, squares, diamonds and asterisks represent the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection, two-dimensional discrete cosine transform and metric multidimensional scaling, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively.

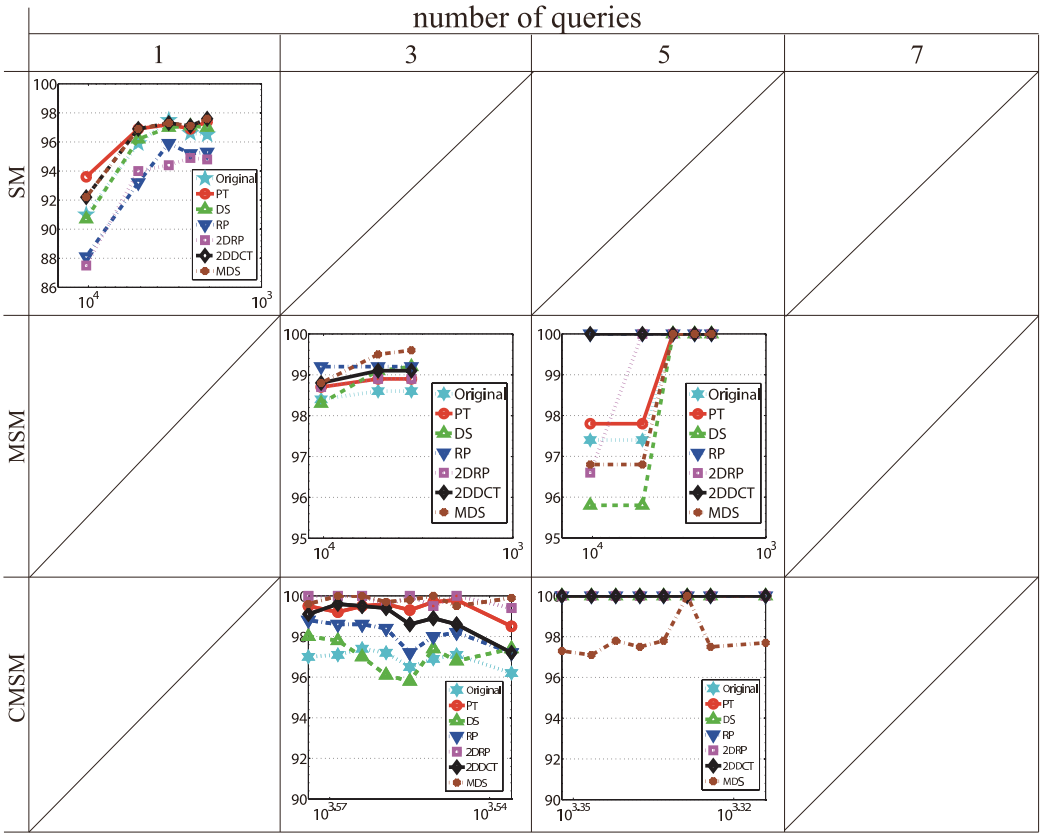


Figure 3.13: Recognition rates for each pair consisting of dimension-reduction method and classification method in the ORL dataset. Stars, circles, upward triangles, downward triangles, squares, diamonds and asterisks represent the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection, two-dimensional discrete cosine transform and metric multidimensional scalng, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively.

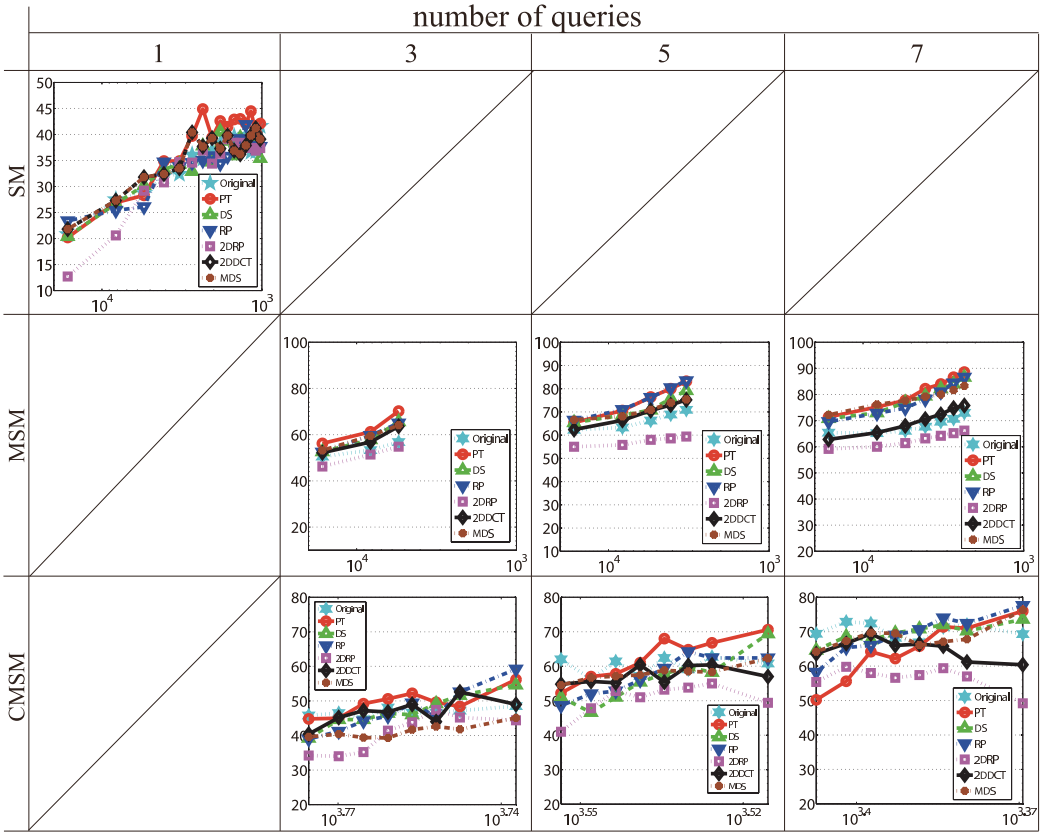


Figure 3.14: Recognition rates for each pair consisting of dimension-reduction method and classification method in the ETH80 dataset. Stars, circles, upward triangles, downward triangles, squares, diamonds and asterisks represent the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection, two-dimensional discrete cosine transform and metric multidimensional scaling, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively.

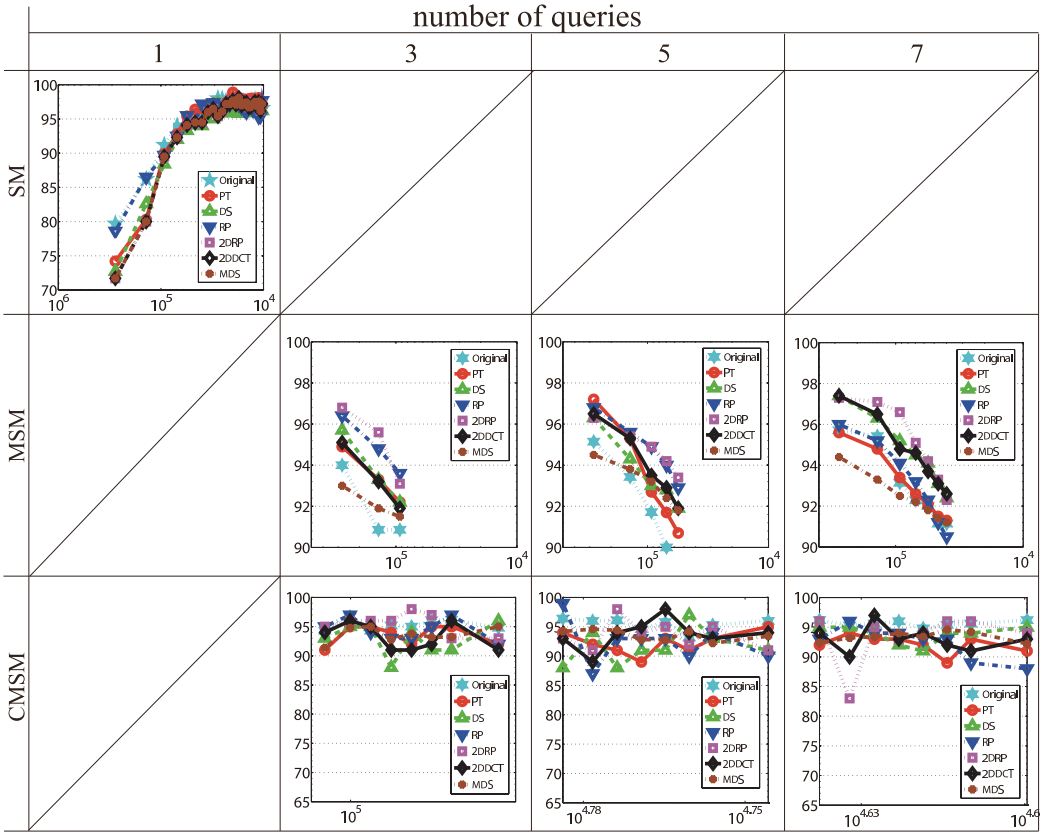


Figure 3.15: Recognition rates for each pair consisting of dimension-reduction method and classification method in the NEC animal dataset. Stars, circles, upward triangles, downward triangles, squares, diamonds and asterisks represent the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection, two-dimensional discrete cosine transform and metric multidimensional scaling, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively.

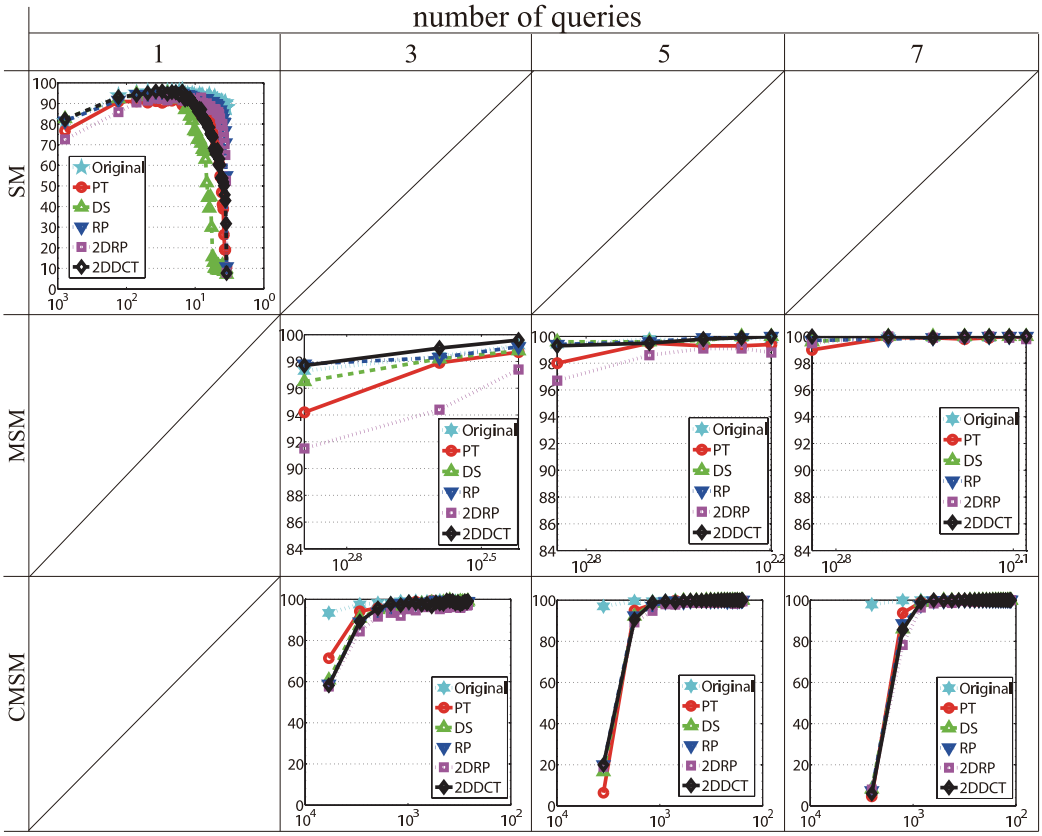


Figure 3.16: Recognition rates for each pair consisting of dimension-reduction method and classification method in the MNIST dataset. Stars, circles, upward triangles, downward triangles, squares and diamonds represent the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection and two-dimensional discrete cosine transform, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively.

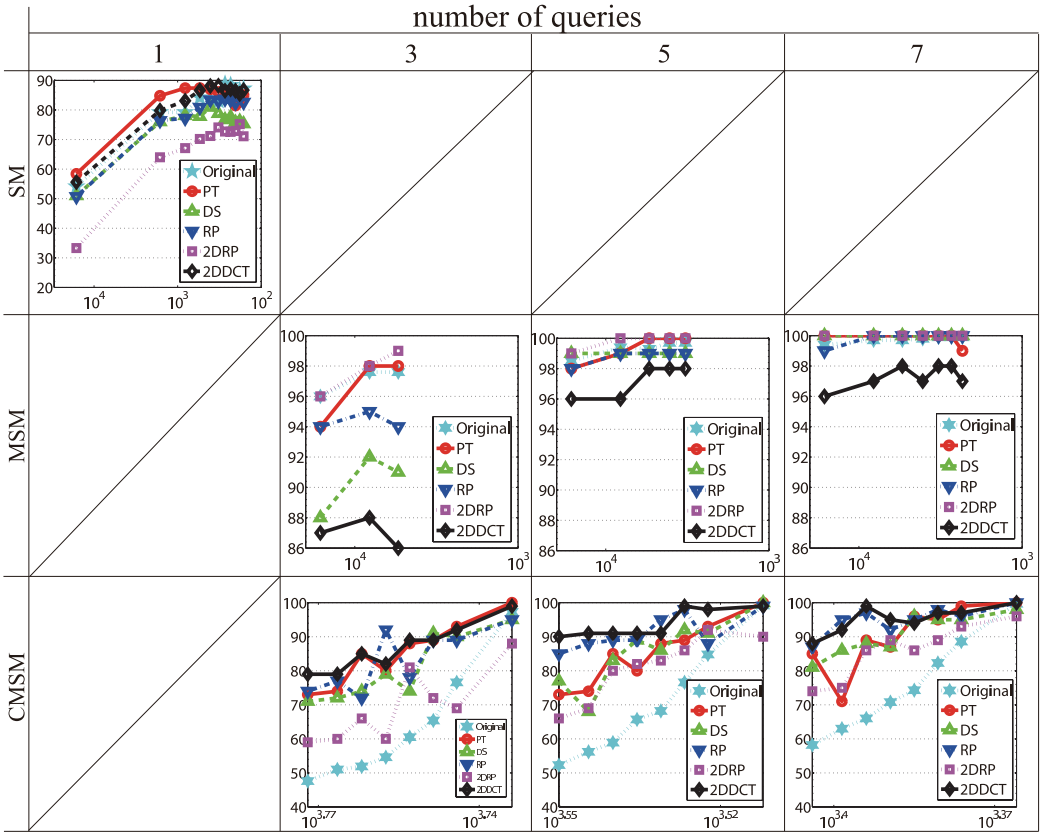


Figure 3.17: Recognition rates for each pair consisting of dimension-reduction method and classification method in the ETL9G dataset. Stars, circles, upward triangles, downward triangles, squares and diamonds represent the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection and two-dimensional discrete cosine transform, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively.

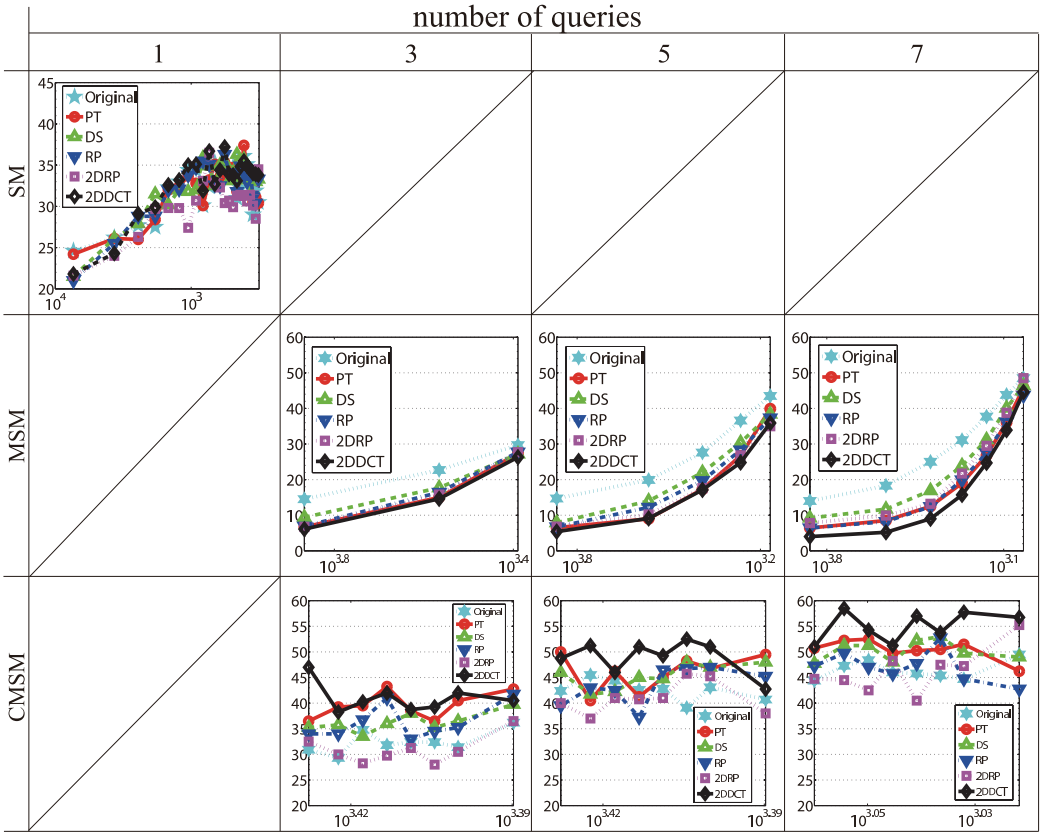


Figure 3.18: Recognition rates for each pair consisting of dimension-reduction method and classification method in the CALTECH101 dataset. Stars, circles, upward triangles, downward triangles, squares and diamonds represent the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection and two-dimensional discrete cosine transform, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively.

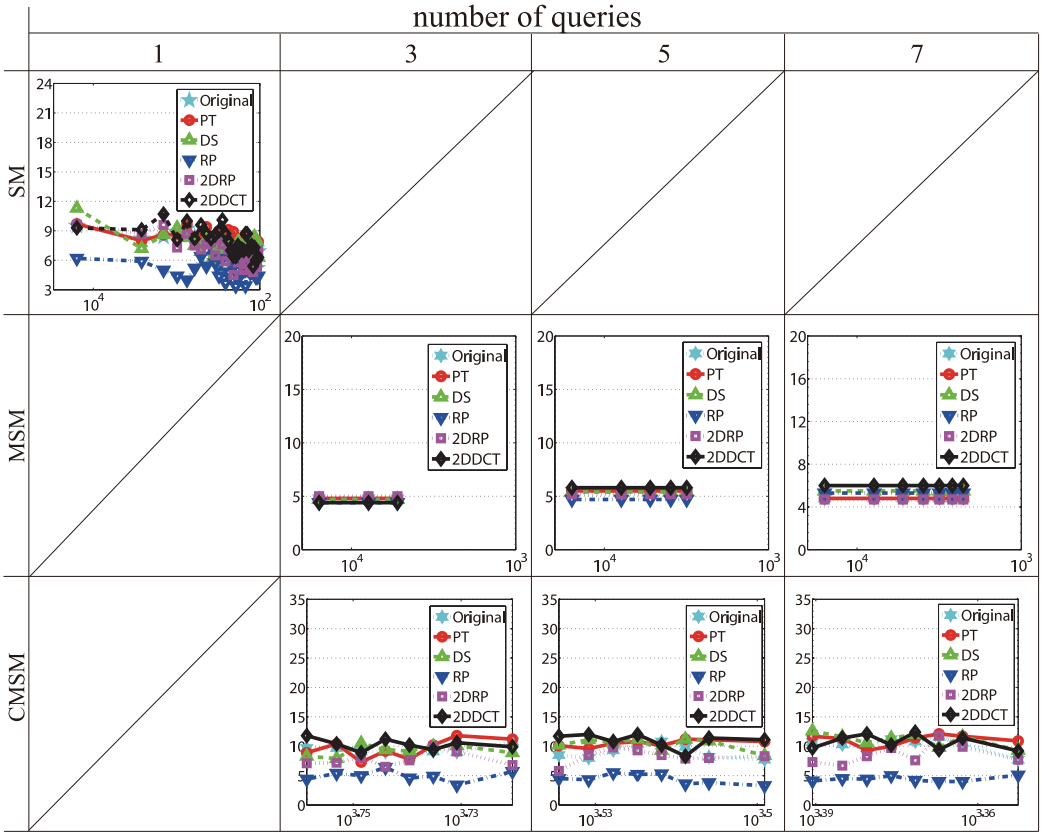
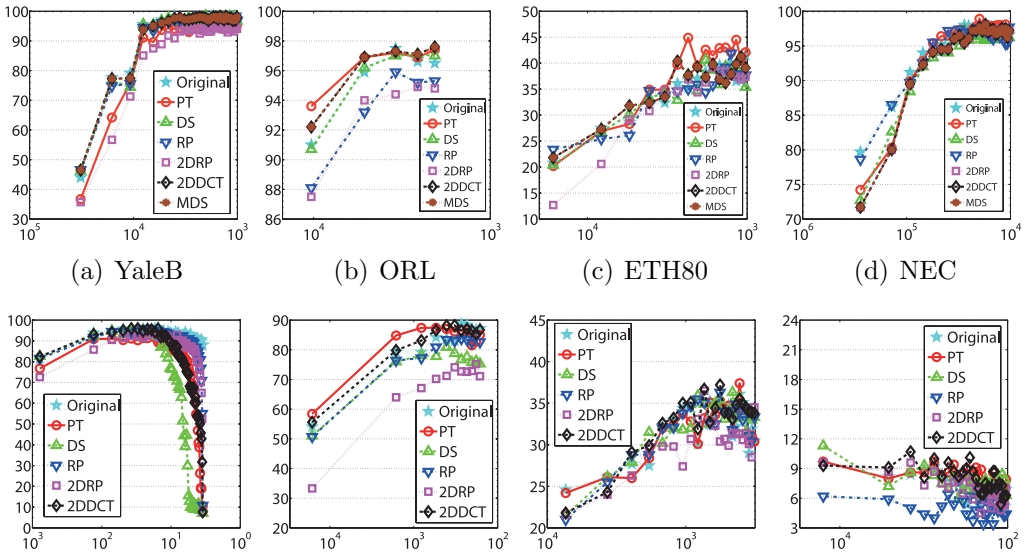


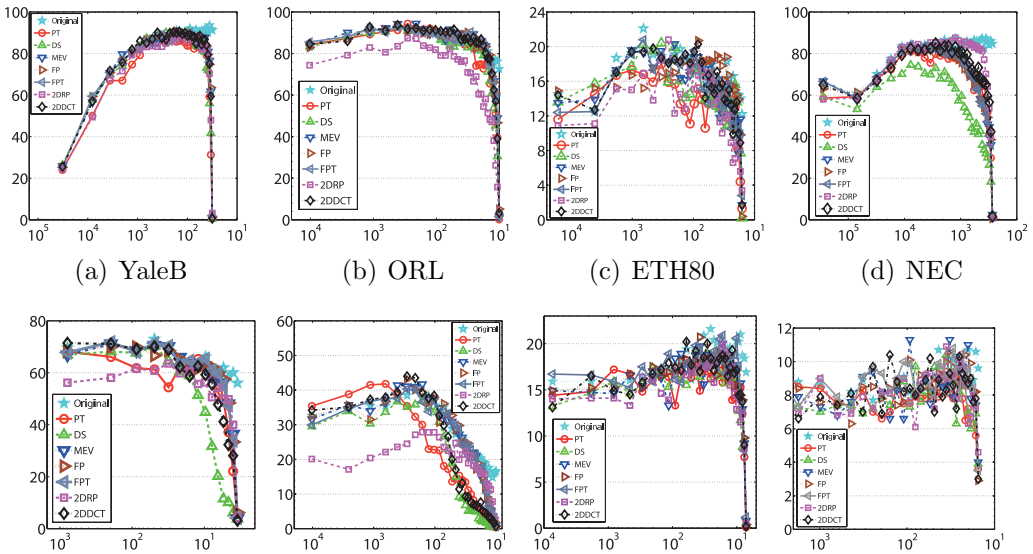
Figure 3.19: Recognition rates for each pair consisting of dimension-reduction method and classification method in the VOC2012 dataset. Stars, circles, upward triangles, downward triangles, squares and diamonds represent the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection and two-dimensional discrete cosine transform, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively.



(a) YaleB (b) ORL (c) ETH80 (d) NEC

(e) MNIST (f) ETL9G (g) CALTECH101 (h) VOC2012

Figure 3.20: Recognition rates for the vector-representation-based subspace method. In (a)-(h), stars, circles, upward triangles, downward triangles, squares and diamonds represent the original dimension, pyramid transform, downsampling, random projection, two-dimensional random projection and two-dimensional discrete transform, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively.



(a) YaleB (b) ORL (c) ETH80 (d) NEC

(e) MNIST (f) ETL9G (g) CALTECH101 (h) VOC2012

Figure 3.21: Recognition rates of the matrix-representation-based tensor subspace method. In (a)-(h), stars, circles, upward triangles, downward triangles, rightward triangles, leftward triangles, squares and diamonds represent the original dimension, pyramid transform, downsampling, marginal eigenvector method, full projection method, full projection truncation method, two-dimensional random projection and two-dimensional discrete transform, respectively. In these figures, the horizontal and vertical axes represent the compression ratio and recognition rate, respectively.

Chapter 4

Recognition of Multilinear Forms

This Chapter is based on three published works. This chapter mainly based on Publication of Journal Paper "1. Pattern Recognition in Multilinear Space and its Applications: Mathematics, Computational Algorithms and Numerical Validations" and Publication of International Conference "1. Approximation of N-Way Principal Component Analysis for Organ Data". The numerical experiments of gait data, volumetric data and sequence of volumetric data are based on the results presented in Publication of Journal Paper "1. Pattern Recognition in Multilinear Space and its Applications: Mathematics, Computational Algorithms and Numerical Validations", Publication of International Conference "2. Classification of Volumetric Data using Multiway Data Analysis" and Publication of International Conference "1. Approximation of N-Way Principal Component Analysis for Organ Data", respectively.

4.1 Recognition Methods for Multilinear Forms

In this chapter, we introduce recognition methods for higher-order tensors. A pattern is assumed to be a square integrable function defined on a finite support in n-dimensional Euclidean space. For planar and volumetric pattern, the dimensions of Euclidean spaces are two and three, respectively. Organs are essentially spatial textures which are functions defined in three-dimensional Euclidean space. Furthermore, for video sequence [120] and volumetric sequence [169], the dimensions of the data spaces are three and four, respectively, since in these applications for planar- and spatio-temporal

Table 4.1: Glossary of abbreviations.

SVD	singular value decomposition
PCA	principal component analysis
TPCA	tensor principal component analysis
MEV	marginal eigenvector
FP	full projection
FPT	full projection truncation
2DPCA	two-dimensional principal component analysis
GPCA	generalised principal component analysis
2DSVD	two-dimensional principal component analysis
2DDCT	two-dimensional discrete cosine transform
ALS	alternating least squares
NDSVD	N-dimensional singular value decomposition
HOSVD	higher-order singular value decomposition
MPCA	multilinear principal component analysis
3DDCT	three-dimensional discrete cosine transform
NDDCT	N-dimensional discrete cosine transform

data are focused to analysis. Moreover, planar multichannel images [103, 51] are also expressed as three-way arrays. For these data, elements of two-dimensional array use an additional axis to express frequencies of elements. Multichannel image pattern recognition has been a central issue in remote sensing of earth and planets [26]. In seismic data analysis, the dimension of the data space is five, since waves stated by a planar source array migrated to planar receiver array are focused to analyse [49].

Table 4.1 summarises the abbreviations that we use in this chapter.

4.2 Related Works

The vector PCA method is a traditional method for data compression. It has been extended to higher-dimensional array data [112, 111]. For example, tensor PCA method constructs a small size tensor using the orthogonal decomposition of a tensor, while the classical PCA (vector PCA) estimates a low-dimensional linear subspace using PCA. Second-order TPCA, which directly decomposes an image matrix, is the tensor PCA method [112, 111] for two-dimensional images. A survey [111] reported that there are three basic projections for a tensor. Second- and third-order TPCA use tensor-to-tensor projections consisting of 1- and 2-mode projections, and 1-, 2- and 3-mode projections for two-dimensional and three-dimensional images, respectively.

For image representation, two-dimensional principal component analysis (2DPCA) [184] has been proposed. However, the projection method in the 2DPCA is not a bilinear form since the 2DPCA uses only the 2-mode projection. The MEV method [131], which is based on both 1- and 2-mode projections, has been proposed for image compression. The 2DSVD [1, 44], which is also based on both 1- and 2-mode projections, has been proposed for image compression as an extension of the SVD [62]. The projections in the MEV method and the 2DSVD are equivalent to the tensor-to-tensor projection for a second-order tensor. This mathematical property implies that the 2DSVD is a special case of the TPCA. However, the compression rate of the 2DSVD is smaller than that of the 2DDCT [1] for the same reconstruction quality. An iterative algorithm [185] for the second-order TPCA has been proposed. This iterative algorithm is referred to as generalised principal component analysis (GPCA). This GPCA method is a two-dimensional version of the iterative algorithm for the SVD [72, 125].

The origin of the TPCA for the third-order tensors was proposed as the decomposition of tensors by Tucker [160]. For the Tucker decomposition of second- and third-order tensors, Kroonenberg and Jeeuw discussed the properties of convergence of alternating-least-squares (ALS) algorithms [95]. In general for Tucker decomposition, orthogonality constraints on decomposed tensors are not required. Cichoki *et al.* imposed that the existence of the constraints is the difference between the TPCA and parallel factor analysis [35]. In ref. [35], in addition to orthogonal constraints, sparse constraints and nonnegative constraints for tensor decomposition are studied. For practical computation in higher TPCA, higher-order singular value decomposition (HOSVD) [100] was formulated. In ref. [100, 99], ALS algorithm for the smaller-size tensor approximation of higher-order tensors was studied. For pattern recognition, there are variants of the TPCA [112, 77, 113, 8]. Multilinear principal component analysis (MPCA) [112] is an iterative algorithm used for TPCA and is the N-dimensional version of the GPCA. This iterative algorithm is the same as the ALS algorithm. Robust MPCA [77] is a robust version of TPCA for image pattern recognition including outliers. Uncorrelated MPCA [113] searches for a tensor-to-vector projection that obtains most of the variation in the original tensorial input by deciding the maximum number of uncorrelated features. Sparse higher-order principal component analysis [8] searches for the minimum number of bases for input tensors by assuming sparsity in tensor decomposition.

For the construction of classifiers, supervised tensor learning frameworks have been proposed [183, 155, 94, 66, 81]. As an extension of linear discriminant analysis, multilinear discriminant analysis has been proposed [183]. As a extension of linear support vector machine to tensors, support tensor

machine has been proposed [155, 94, 66]. These methods are basically two-class classifiers. Itoh *et al.* proposed As extension of the subspace method [76, 176, 175, 130]. the two-dimensional tensor subspace method (2DTSM) [81], which measures the similarity between an input image and each tensor subspace of a class, for image pattern recognition. The 2DTSM adopts the MEV method to construct each tensor subspace of a class. We extend the 2DTSM for use with three-dimensional images in this Chapter.

4.3 Tensor Subspace of Categories

Setting $\{\boldsymbol{U}_k^{(j)}\}_{j=1}^N$ to be orthogonal matrices of a tensor projection for Nth-order tensors, we have a tensor subspace spanned by $\{\boldsymbol{U}_k^{(j)}\}_{j=1}^N$ for kth category. Therefore, we can define a tensor subspace of a category by

$$\mathcal{C}_k = \{\mathcal{X} \mid \mathcal{X} \times_1 \boldsymbol{U}_k^{(1)\top} \times_2 \boldsymbol{U}_k^{(2)\top} \cdots \times_n \boldsymbol{U}_k^{(N)} = \mathcal{X}\}. \tag{4.1}$$

Since a pattern represented by tensors contains perturbation, we define kth category by

$$\mathcal{C}_k(\delta) = \{\mathcal{X} \mid \|\mathcal{X} \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \cdots \times_n \boldsymbol{U}^{(N)} - \mathcal{X}\|_{\mathrm{F}} \ll \delta\}, \tag{4.2}$$

where a positive constant δ is the bound for a small perturbation to a pattern. Therefore, by defining similarity and dissimilarity between a tensor subspace and query, we can construct tensor-subspace-based classifiers that are robust and stable against small perturbations to patterns.

4.4 Tensor Subspace Method

As an extension of the subspace method [76, 177] for N-way data, we introduce a new linear tensor subspace method for Nth-order tensors. This method is a N-dimensional version of the 2DTSM [81].

For a Nth-order tensor $\mathcal{X}$, we set $\boldsymbol{U}^{(j)},\ j = 1, 2, \ldots, N$, to be projection matrices of the tensor-to-tensor projection of $\mathcal{X}$ to $\mathcal{Y}$. For a collection of normalised tensors $\{\mathcal{X}_i\}_{i=1}^M$, such that $\mathcal{X}_i \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, $\|\mathcal{X}_i\|_{\mathrm{F}} = 1$ and $\mathrm{E}(\mathcal{X}_i) = 0$, the solutions of

$$\{U^{(j)}\}_{j=1}^N = \arg\max \mathrm{E}\left(\|\mathcal{X} \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \cdots \times_N \boldsymbol{U}^{(N)\top}\|_{\mathrm{F}} / \|\mathcal{X}_i\|_{\mathrm{F}}\right) \tag{4.3}$$

with respect to $\boldsymbol{U}^{(j)\top}\boldsymbol{U}^{(j)} = \boldsymbol{I}$ for $j = 1, 2, \ldots, N$ define a multilinear subspace that approximates $\{\mathcal{X}_i\}_{i=1}^M$. Therefore, using projection matrices

$\{\boldsymbol{U}_k^{(j)}\}_{j=1}^N$ obtained as the solutions of eq. (4.3) for kth category, if a query tensor $\mathcal{G}$ satisfies the condition

$$\arg\left(\max_l \|\mathcal{G} \times_1 \boldsymbol{U}_l^{(1)\top} \times_2 \boldsymbol{U}_l^{(2)\top} \cdots \times_N \boldsymbol{U}_l^{(N)\top}\|_{\mathrm{F}} / \|\mathcal{G}\|_{\mathrm{F}}\right) = \{\boldsymbol{U}_k^{(j)}\}_{j=1}^N, \quad (4.4)$$

we conclude that $\mathcal{G} \in \mathcal{C}_k$, $k, l = 1, 2, \ldots, N_{\mathcal{C}}$, where $\mathcal{C}_k$ and $N_{\mathcal{C}}$ are the tensor subspace of kth category and the number of categories, respectively.

4.5 Tensor Subspace of Queries

We have a collection of query tensors $\{\mathcal{G}_{i'}\}_{i'=1}^{M'}$ normalised by $\mathcal{G}_{i'}/\|\mathcal{G}_{i'}\|_{\mathrm{F}}$. We assume that these queries belong to the same category. Then, using multiway PCA for a collection of queries $\{\mathcal{G}_{i'}\}_{i'=1}^{M'}$, we obtain orthogonal matrices $\{\boldsymbol{V}^{(j)}\}_{j=1}^N$, which are given by eq. (4.3) for each mode. The column vectors of orthogonal matrices $\{\boldsymbol{V}^{(j)}\}_{j=1}^N$ are eigenvectors of a set of unfolded tensors for each mode. We note that even if we have only one query $\mathcal{G}$ we can obtain eigenvectors of j-mode by the multiway PCA, since j-mode unfolding gives a set of column vectors of mode-j as shown in Figs. 2.2 and 2.5. That is, for a query $\mathcal{G}$, we have orthogonal matrices by the decomposition [100]

$$\mathcal{G} = \mathcal{A} \times_1 \boldsymbol{V}^{(1)} \times_2 \boldsymbol{V}^{(2)} \cdots \times_N \boldsymbol{V}^{(N)}. \quad (4.5)$$

4.6 Mutual Tensor Subspace Method

As the extension of mutual subspace method for vector data [116], we define a classifier for two tensor subspaces. For each of N_{C} categories of tensor data, we set a collection of normalised Nth-order tensors $\{\mathcal{X}_i\}_{i=1}^M$, such that $\mathcal{X}_i \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, $\|\mathcal{X}_i\|_{\mathrm{F}} = 1$ and $\mathrm{E}(\mathcal{X}_i) = 0$. For the kth category in N_{C} categories, we have an orthogonal matrices $\{\boldsymbol{U}_k^{(j)}\}_{j=1}^N$, which satisfy eq. (4.3). The orthogonal matrices for kth category span a tensor subspaces $\mathcal{C}_k$ of the category .

We have a collection of query tensors $\{\mathcal{G}_{i'}\}_{i'=1}^{M'}$ normalised by $\mathcal{G}_{i'}/\|\mathcal{G}_{i'}\|_{\mathrm{F}}$. We assume that these queries belong to the same category. Then, using multiway PCA for a collection of queries $\{\mathcal{G}_{i'}\}_{i'=1}^{M'}$, we obtain orthogonal matrices $\{\boldsymbol{V}^{(j)}\}_{j=1}^N$. The obtained eigenvectors of all modes span a tensor subspace $\mathcal{C}_{\mathrm{q}}$ for queries. For the classification of queries, we measure the dissimilarity between a category subspace $\mathcal{C}_k$ and a query subspace $\mathcal{C}_{\mathrm{q}}$. Since $\mathcal{C}_k$ and $\mathcal{C}_q$ represent patterns with perturbations, we can robustly recognise queries against the pattern perturbations by measuring the dissimilarity between two tensor subspace.

Using orthogonal matrices $\{\boldsymbol{U}_k^{(j)}\}_{j=1}^N$ for kth category, we have a projected tensor in a category subspace $\mathcal{C}_k$ by

$$\mathcal{A}_{i'} = \mathcal{G}_{i'} \times_1 \boldsymbol{U}_k^{(1)\top} \times_2 \boldsymbol{U}_k^{(2)\top} \cdots \times_N \boldsymbol{U}_k^{(N)\top}. \tag{4.6}$$

Furthermore, using orthogonal matrices $\{\boldsymbol{V}^{(j)}\}_{j=1}^N$, we have a projected tensor in a query subspace $\mathcal{C}_{\mathrm{q}}$ by

$$\mathcal{B}_{i'} = \mathcal{G}_{i'} \times_1 \boldsymbol{V}^{(1)\top} \times_2 \boldsymbol{V}^{(2)\top} \cdots \times_N \boldsymbol{V}^{(N)\top}. \tag{4.7}$$

For a tensor subspaces $\mathcal{C}_k$ and $\mathcal{C}_{\mathrm{q}}$, we define the dissimilarity of subspaces $d(\mathcal{C}_k, \mathcal{C}_{\mathrm{q}})$ by

$$\mathrm{E}\left(\|\mathcal{A}_{i'} \times_1 \boldsymbol{P}\boldsymbol{U}_k^{(1)} \cdots \times_N \boldsymbol{P}\boldsymbol{U}_k^{(N)} - \mathcal{B}_{i'} \times_1 \boldsymbol{P}\boldsymbol{V}^{(1)} \cdots \times_N \boldsymbol{P}\boldsymbol{V}^{(N)}\|_{\mathrm{F}}^2\right), \tag{4.8}$$

where a projection matrix $\boldsymbol{P}$ selects bases for each mode of tensors. Therefore, using the dissimilarity given by eq. (4.8), if queries $\{\mathcal{G}_{i'}\}_{l=1}^{M'}$ satisfy the condition

$$\arg\left(\min_l d(\mathcal{C}_l, \mathcal{C}_{\mathrm{q}})\right) = \mathcal{C}_k, \tag{4.9}$$

we conclude that $\{\mathcal{G}_{i'}\}_{i'=1}^{M'} \in \mathcal{C}_k(\delta)$ for $k, l = 1, 2, \ldots, N_{\mathrm{C}}$.

4.7 Geometry of Multilinear Subspace

For the simplification of discussion, we consider second-order tensors, that is, two-dimensional images. For instances $i = 1, 2, \ldots, N$ $t_1, t_2, \ldots, t_N$, we have a set $X = \{\boldsymbol{X}_i\}_{i=1}^N$ of two-dimensional digital images. For the set $X = \{\boldsymbol{X}_i\}_{i=1}^N$, we compute orthogonal matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ that minimise

$$J(\boldsymbol{U}, \boldsymbol{V}) = \underset{i}{E}\|\boldsymbol{X}_i - \boldsymbol{U}\boldsymbol{\Sigma}_i\boldsymbol{V}^\top\|_{\mathrm{F}}^2, \tag{4.10}$$

where $\boldsymbol{\Sigma}_i$ is a coefficient matrix. If we apply tensor principal component analysis to each image $\boldsymbol{X}_i$, we have orthogonal matrices $\boldsymbol{U}_i$ and $\boldsymbol{V}_i$ that minimise

$$J(\boldsymbol{U}_i, \boldsymbol{V}_i) = \|\boldsymbol{X}_i - \boldsymbol{U}_i\boldsymbol{\Sigma}_i\boldsymbol{V}_i^\top\|_{\mathrm{F}}^2, \tag{4.11}$$

where $\boldsymbol{\Sigma}$ is a diagonal matrix.

For $\boldsymbol{X}_i$, $\boldsymbol{X}_{i+1} \in X$, we have pairs of orthogonal matrices $\langle \boldsymbol{U}_i, \boldsymbol{V}_i \rangle$ and $\langle \boldsymbol{U}_{i+1}, \boldsymbol{V}_{i+1} \rangle$, respectively. For these two pairs, we define rotation matrices $\boldsymbol{R}_i^{(1)}$ and $\boldsymbol{R}_i^{(2)}$ by

$$\boldsymbol{U}_{i+1} = \boldsymbol{R}_i^{(1)}\boldsymbol{U}_i, \tag{4.12}$$

$$\boldsymbol{V}_{i+1} = \boldsymbol{R}_i^{(2)}\boldsymbol{V}_i, \tag{4.13}$$

for 1- and 2-mode eigenvectors, respectively. $\boldsymbol{R}_i^{(1)}$ and $\boldsymbol{R}^2$ are given by

$$\boldsymbol{R}_i^{(1)} = \boldsymbol{U}_{i+1}\boldsymbol{U}_i^\top, \tag{4.14}$$

$$\boldsymbol{R}_i^{(2)} = \boldsymbol{V}_{i+1}\boldsymbol{V}_i^\top. \tag{4.15}$$

If orthogonal matrices satisfy $\boldsymbol{U}_i = \boldsymbol{U}_{i+1}$ and $\boldsymbol{V}_i = \boldsymbol{V}_{i+1}$, then relations $\boldsymbol{R}^{(1)} = \boldsymbol{R}^{(2)} = \boldsymbol{I}$ hold.

For a pair of images $f(x, y)$ and $g(x, y)$, setting

$$p(x, y) = \frac{f(x, y)}{F},\ F = \int\int_{\mathbf{R}^2} f(x, y)dxdy, \tag{4.16}$$

$$q(x, y) = \frac{g(x, y)}{G},\ F = \int\int_{\mathbf{R}^2} g(x, y)dxdy \tag{4.17}$$

The transportation between $f(x, y)$ and $g(x, y)$ is computed

$$T(f, g) = \min_c \int\int_{\mathbf{R}^2}\int\int_{\mathbf{R}^2} |p(x, y) - q(x', y')|c(x, y; x'y')dxdydx'dy', \tag{4.18}$$

where $\int\int_{\mathbf{R}^2} c(x, y; x'y')dx'dy' = p(x, y)$ and $\int\int_{\mathbf{R}^2} c(x, y; x'y')dxdy = q(x', y')$. If $f(x, y)$ and $g(x, y)$ are sampled on the $N \times N$ grid, the minimisation of the discrete version of eq. (4.18)

$$T(f, g) = \min_{c_{iji'j'}} \sum_{ij=1}^{N}\sum_{i'j'}^{N} |p_{ij} - q_{i'j'}|c_{iji'j'}, \tag{4.19}$$

where $\sum_{i'j'} c_{iji'j'} = p_{ij}$ and $\sum_{ij} c_{iji'j'} = q_{i'j'}$, is achieved using linear programming for $(N \times N)^2$ dimensional vectors.

For two bases $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$, we define geodes tic distance $d(\cdot, \cdot)$ by

$$d(\boldsymbol{u}_i, \boldsymbol{u}_j) = \begin{cases} \cos^{-1}(\boldsymbol{u}_i^\top \boldsymbol{u}_j), & \text{if } 0 \leq \boldsymbol{u}_i^\top \boldsymbol{u}_j \leq 1 \\ \cos^{-1}(|\boldsymbol{u}_i^\top \boldsymbol{u}_j|), & \text{if } -1 \leq \boldsymbol{u}_i^\top \boldsymbol{u}_j < 0. \end{cases} \tag{4.20}$$

We set $\boldsymbol{U}^k = [\boldsymbol{u}_1^k, \ldots, \boldsymbol{u}_N^k]$, $\boldsymbol{V}^k = [\boldsymbol{v}_1^k, \ldots, \boldsymbol{v}_N^k]$ and $\boldsymbol{U}^{k+1} = [\boldsymbol{u}_1^{k+1}, \ldots, \boldsymbol{u}_N^{k+1}]$, $\boldsymbol{V}^{k+1} = [\boldsymbol{v}_1^{k+1}, \ldots, \boldsymbol{v}_N^{k+1}]$. Setting $d_{ij}^{(1)} = d(\boldsymbol{u}_i^{k+1}, \boldsymbol{u}_j^k)$ and $d_{ij}^{(2)} = d(\boldsymbol{v}_i^{k+1}, \boldsymbol{v}_j^k)$, and by computing tensor principal component analysis for $\boldsymbol{X}^k, \boldsymbol{X}^{k+1}$ as a preprocessing, we approximate the transportation problem in eq. (4.19) to the minimisation of

$$d(X_{k+1}, X_k) = \min_{c_{ij}^{(1)}} \sum_{i,j=1}^{N} d_{ij}^{(1)} c_{ij}^{(1)} + \min_{c_{ij}^{(2)}} \sum_{i,j=1}^{N} d_{ij}^{(2)} c_{ij}^{(2)}. \tag{4.21}$$

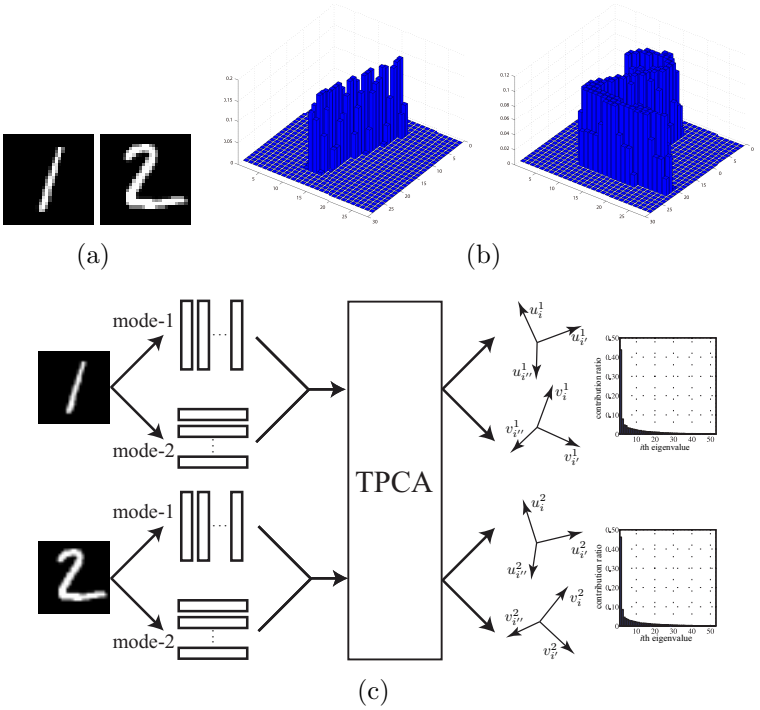


Figure 4.1: Mathematical properties of expression of images for computation of the distances between images. (a) Gray-scale images. We generally use L_2-norm for them as a distance. representation. (b) Probabilistic distribution of gray values in images. Images are represented by probabilistic distributions by normalised as there L_2-norms to be 1. Wasserstein distance is defined by the sum of transportation costs between two probabilistic distributions. (c) Decomposition of images by tensor principal component analysis. Images are decomposed to eigenvalues and eigenvectors. Wasserstein distance is defined by the sum of transportation costs for contribution ratios of eigenvalues. In the transportation, an angle between bases is adopted as a cost for transportation.

For this minimisation problem, we give constraint conditions

$$\sum_j c_{ij}^{(1)} = \sum_j c_{ij}^{(2)} = \lambda_i^{k+1} / \sum_{i=1}^{N}, \lambda_i^{k+1} \tag{4.22}$$

$$\sum_i c_{ij}^{(1)} = \sum_i c_{ij}^{(2)} = \lambda_j^{k} / \sum_{j=1}^{N} \lambda_j^{k}. \tag{4.23}$$

where λ_j^k and λ_i^{k+1} are eigenvalues for $\boldsymbol{X}^k$ and $\boldsymbol{X}^{k+1}$, respectively. Therefore, the problem is transformed to linear programming for $N \times N$-dimensional vectors. Figure 4.1 summarises mathematical properties of image expressions. Figure 4.2 illustrates Wasserstein distance based on mode-1 and -2 bases for second-order tensor. If the angles between $\boldsymbol{u}_i^{k+1}$ and $\boldsymbol{u}_j^k$ and $\boldsymbol{v}_i^{k+1}$ and $\boldsymbol{v}_j^k$ are small, the distance defined in eq. (4.21) is approximated by

$$d(\boldsymbol{X}_{k+1}, \boldsymbol{X}_k) = \sum_{i,j}^{N} (d_{ij}^{(1)} + d_{ij}^{(2)}). \tag{4.24}$$

We can apply the distance defined in eq. (4.21) to bases of two tensor subspace by replacing constraint conditions. For two sets $\{\boldsymbol{X}_{1,i}\}_{i=1}^N$ and $\{\boldsymbol{X}_{2,i}\}_{i=1}^N$, we compute pairs of orthogonal matrices $\boldsymbol{U}_1, \boldsymbol{V}_1$ and $\boldsymbol{U}_1, \boldsymbol{V}_2$ as minimisation of eq. (4.10), respectively.$\{\lambda_{2,i}^{|}(1)\}_{i=1}^N$, $\{\lambda_{2,i}^{|}(2)\}_{i=1}^N$, These two pairs span tensor subspaces X_1 and X_2 for two sets $\{\boldsymbol{X}_{1,i}\}_{i=1}^N$ and $\{\boldsymbol{X}_{2,i}\}_{i=1}^N$, respectively. We set $\boldsymbol{U}_1 = [\boldsymbol{u}_{1,1}, \ldots, \boldsymbol{u}_{1,N}]$, $\boldsymbol{V}_1 = [\boldsymbol{v}_{1,1}, \ldots, \boldsymbol{v}_{1,N}]$ and $\boldsymbol{U}_2 = [\boldsymbol{u}_{2,1}, \ldots, \boldsymbol{u}_{2,N}]$, $\boldsymbol{V}_2 = [\boldsymbol{v}_{2,1}, \ldots, \boldsymbol{v}_{2,N}]$. Eigenvalues $\lambda_{1,i}^{(1)}$ and $\lambda_{1,i}^{(2)}$ are correspond to $\boldsymbol{u}_{1,i}$ and $\boldsymbol{v}_{1,i}$, respectively. Eigenvalues $\lambda_{2,i}^{(1)}$ and $\lambda_{2,i}^{(2)}$ are correspond to $\boldsymbol{u}_{2,i}$ and $\boldsymbol{v}_{2,i}$, respectively. Setting $d_{ij}^{(1)} = d(\boldsymbol{u}_{1,i}, \boldsymbol{u}_{2,j})$ and $d_{ij}^{(2)} = d(\boldsymbol{v}_{1,i}, \boldsymbol{v}_{2,j})$, we define distance between two subspaces X_1 and X_2 as transpiration problem by

$$d(X_1, X_2) = \min_{c_{ij}^{(1)}} \sum_{i,j=1}^{N} d_{ij}^{(1)} c_{ij}^{(1)} + \min_{c_{ij}^{(2)}} \sum_{i,j=1}^{N} d_{ij}^{(2)} c_{ij}^{(2)}, \tag{4.25}$$

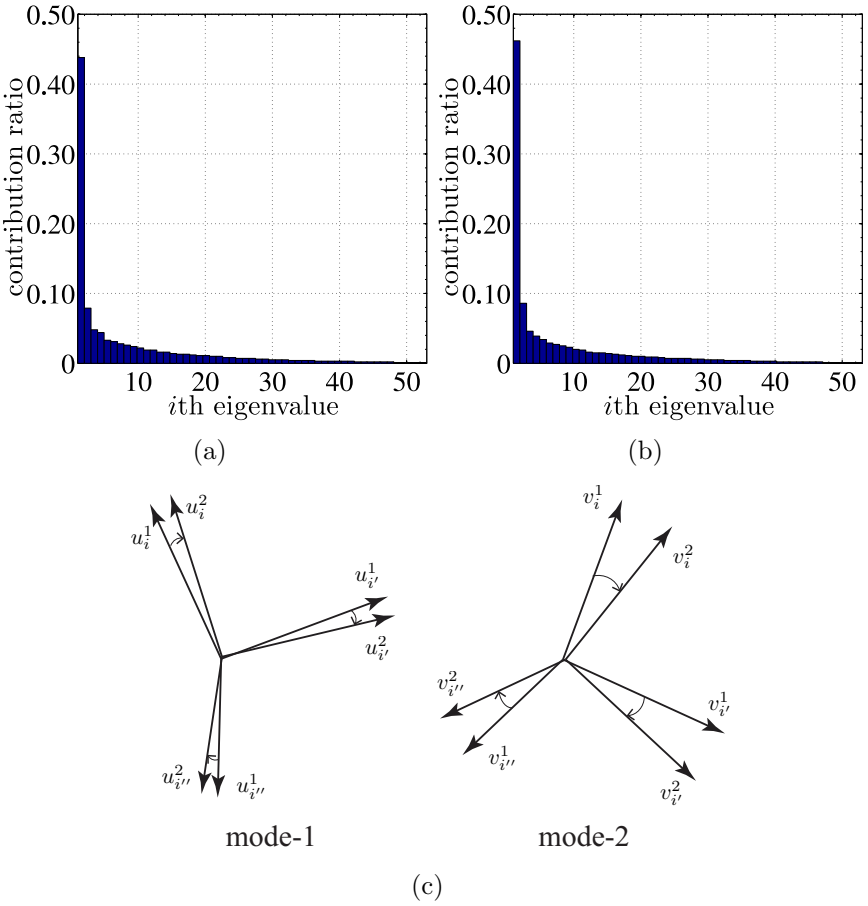


Figure 4.2: The Wasserstein distance between tensor subspaces for second-order tensors. Tensor subspaces are obtained by tensor principal component analysis for each given image. Note that a tensor subspace is obtained from an image. For these subspaces, transportation between contribution ratios of eigenvalues is considered. (a) and (b) show contribution ratios of eigenvalues obtained by singular value decomposition for different two images. An angle between bases are computed as a transportation cost between eigenvalues. Wasserstein distance is the result of the minimisation of total transportation cost among eigenvalues for two tensor subspaces.

where we define constraints conditions

$$\sum_j c_{ij}^{(1)} = \lambda_i^{(1)} / \sum_{i=1}^{N} \lambda_i^{(1)}, \tag{4.26}$$

$$\sum_j c_{ij}^{(2)} = \lambda_i^{(2)} / \sum_{i=1}^{N} \lambda_i^{(2)}, \tag{4.27}$$

$$\sum_i c_{ij}^{(1)} = \lambda_j^{(1)} / \sum_{j=1}^{N} \lambda_j^{(1)}, \tag{4.28}$$

$$\sum_i c_{ij}^{(2)} = \lambda_j^{(2)} / \sum_{j=1}^{N} \lambda_j^{(2)}. \tag{4.29}$$

A geodesic distance is a unit cost for transportation of cumulative contribution ratio for each eigenvector.

Furthermore, we can define Wasserstein distance among Mth-order tensors. For two Mth-order tensors $\mathcal{X}_1,\ \mathcal{X}_2 \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$, using higher-order singular value decomposition or tensor principal component analysis, we have decompositions

$$\mathcal{X}_1 = \mathcal{Y}_1 \times \boldsymbol{U}_1^{(1)} \times \boldsymbol{U}_1^{(2)} \times \cdots \times \boldsymbol{U}_1^{(M)}, \tag{4.30}$$

$$\mathcal{X}_2 = \mathcal{Y}_2 \times \boldsymbol{U}_2^{(1)} \times \boldsymbol{U}_2^{(2)} \times \cdots \times \boldsymbol{U}_2^{(M)}. \tag{4.31}$$

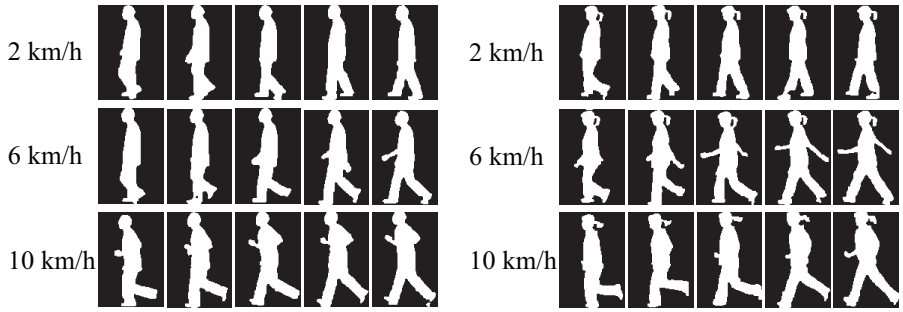
As the results of these higher-order singular value decomposition, we have sets of orthogonal matrices $\{\boldsymbol{U}_1^{(m)} | \boldsymbol{U}_1^{(m)} = [\boldsymbol{u}_{(1,1)}^{(m)} \dots \boldsymbol{u}_{(1,I_m)}^{(m)}],\ m = 1, 2, \dots, M\}$ and $\{\boldsymbol{U}_2^{(m)} | \boldsymbol{U}_2^{(m)} = [\boldsymbol{u}_{(2,1)}^{(m)} \dots \boldsymbol{u}_{(2,I_m)}^{(m)}],\ m = 1, 2, \dots, M\}$ for each mode. In these decompositions, each base $\boldsymbol{u}_{k,l}^{(m)}$ correspond to eigenvalue $\lambda^{(m)_{k,l}}$ for $k = 1, 2$ and $l = 1, 2, \dots, I_m$. Using these bases of orthogonal matrices, we define the Wasserstein distances between two Mth-order tensors by

$$d(\mathcal{X}_1, \mathcal{X}_2) = \min_{c_{ij}^{(1)}} \sum_{i,j=1}^{I_1} d_{ij}^{(1)} c_{ij}^{(1)} + \min_{c_{ij}^{(2)}} \sum_{i,j=1}^{I_2} d_{ij}^{(2)} c_{ij}^{(2)} + \cdots + \min_{c_{ij}^{(M)}} \sum_{i,j=1}^{I_M} d_{ij}^{(M)} c_{ij}^{(M)}, \tag{4.32}$$

where we set constraints conditions

$$\sum_j c_{ij}^{(m)} = \lambda_i^{(m)} / \sum_{i=1}^{I_m} \lambda_i^{(m)} \tag{4.33}$$

$$\sum_i c_{ij}^{(m)} = \lambda_j^{(m)} / \sum_{j=1}^{I_m} \lambda_j^{(m)}. \tag{4.34}$$



(a) Person ♯001 (b) Person ♯128

Figure 4.3: Examples of sequences of silhouette images. The figures are gait images whose pixel values are 0 or 255. The figure illustrate the 1st, 21st, 41st, 61st, 81st silhouette images of sequences from a person walking at different speeds. Each sequence consists of 90 silhouette images of four steps. For each sequence, we manually selected the start and finish of the sequence from the original OU-ISIR treadmill dataset. Each sequence is obtained by resampling of the selected sequence with linear interpolation, where the linear interpolation is only used for mode 3.

We can use this Wasserstein distance to compute the distance between two tensor subspaces for Mth-order tensors.

4.8 Experiments

4.8.1 Gait Patterns

To validate the relation between the HOSVD and the 3DDCT, we compute recognition rates using the OU-ISIR dataset [120]. Figure 4.3 shows examples of sequences of silhouette images from two different categories in the OU-ISIR dataset. Table 4.2 summarises the sizes of tensors of the two datasets. For the compression of the silhouette-image sequences, we use the HOSVD and the 3DDCT. For the practical computation of the HOSVD, we use the iterative method described in Algorithm 1 [112]. If we set the number of iterations to 0 in Algorithm 1, we have the three-dimensional version of the MEV method. If we set the number of bases to the size of the original tensors in Algorithm 1, we call the method full projection (FP). If we set the number of bases to less than the size of the original tensors in Algorithm 1, we call the method full projection truncation (FPT).

Firstly, we observe the properties of the iterative method. Using se-

Table 4.2: Sizes and number of tensors of the resampled OU-ISIR.♯class and ♯data/class represent the number of classes and the number of data in each class, respectively. The tensor size is the size of the dataset before dimension reduction. The reduced tensor size is the size of the tensor after dimension reduction. We set $d \in \{32, 16, 8\}$ for the size in the dimension reduction.

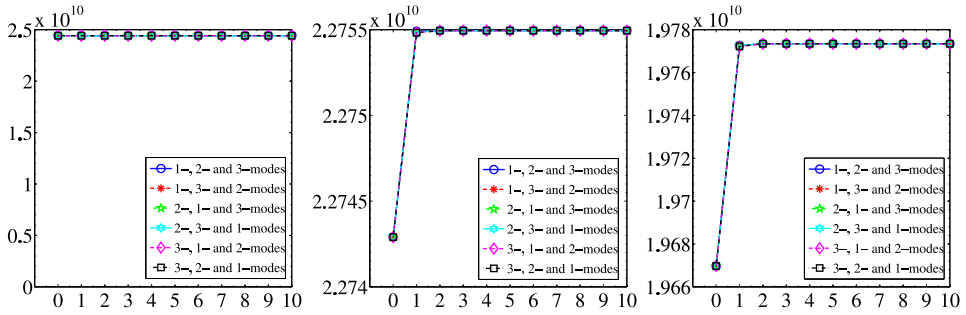
	♯class	♯data /class	tensor size	reduced tensor size
OU-ISIR	34	9	$128 \times 88 \times 90$	$d \times d \times d$

quences of silhouette images from a category, we compute the sum of the energies of projected tensors after k iterations and the CCR of the eigenvalues for the three modes. For the computation of tensor products, we select different orders of selection of the modes. Since there are three modes, we have $3! = 6$ orders. For FPT, we set the numbers of bases for each mode to $64 \times 64 \times 64$ and $32 \times 32 \times 32$. Figure 4.4 shows the sum of the energies Ψ_k after every 10 iterations for the FP and FPT with the six different orders of computation of the tensor projection. Figure 4.5 shows the CCRs of the three modes in the FP for the different orders of computation of the tensor projections. Figure 4.6 summarises the CCRs of the three modes for projections to the three different sizes. Figure 4.7 shows the CCRs of each mode for FP and FPT.

In Fig. 4.4, the iterations do not significantly affect the sum of the energies of the projected tensors. According to both Figs. 4.4 and 4.5, changing the order of computation of the tensor projections does not give different results. In Fig. 4.6, the CCRs of the three decompositions are almost coincident in the three modes. Figure 4.7 shows that the eigenvalues obtained by FP and FPT are not coincident. The decomposition of tensors for the FPT gives larger eigenvalues with a smaller number of bases than those obtained by FP.

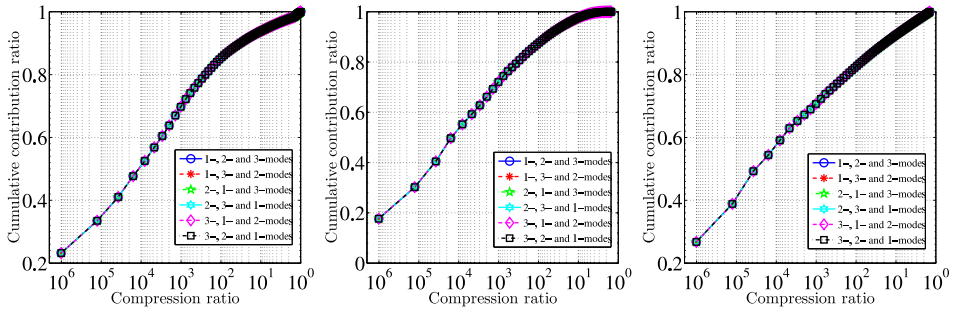
Next, we compute the CCRs of the eigenvalues obtained by 10 iterations of Algorithm 1 for compressed tensors. For the compression of the tensors from $128 \times 88 \times 90$ to $32 \times 32 \times 32$, we adopt the FP, the FPT and the 3DDCT. Figure 4.8 shows the CCRs of each mode for the three types of compressed tensor. The tensors compressed by the 3DDCT give larger eigenvalues than those compressed by the FP and the FPT with a smaller number of bases. The FP and the FPT gives the same CCRs for each mode.

Thirdly, we compute the recognition rate of the sequences of silhouette images by the TSM. In this validation, we use the original sizes of the tensors and compressed tensors for comparison. For the compression, we adopt the



(a) $128 \times 88 \times 90$ (b) $64 \times 64 \times 64$ (c) $32 \times 32 \times 32$

Figure 4.4: Convergence of iteration described in Algorithm 1. (a)-(c) show the sum of energies Ψ_k in each iteration for the given numbers of bases of $128 \times 88 \times 90$, $64 \times 64 \times 64$ and $32 \times 32 \times 32$, respectively. In the (a)-(c), horizontal and vertical axes represent the number of iterations and Ψ_k, respectively. For the computation of the tensor projections using Algorithm 1, we adopt the six orders of selection of the modes, where the legends in the figures summarises the six orders.



(a) mode 1 (b) mode 2 (c) mode 3

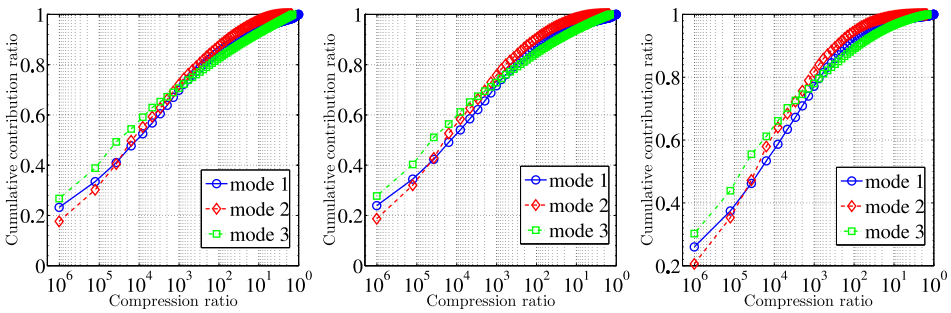
Figure 4.5: Cumulative contribution ratio of eigenvalues obtained by 10 iterations using Algorithm 1. (a)-(c) show the cumulative contribution ratios for the 1-, 2- and 3-modes, respectively. Here, the given number of bases is $128 \times 88 \times 90$ for Algorithm 1. For the computation of the tensor projections using Algorithm 1, we adopt six orders of the selection of modes, where the legends in the figures summarises the six orders. The horizontal and vertical axes represent the compression ratio and cumulative contribution ratio, respectively. For the original size $D = 128 \times 88 \times 90$ and the reduced size $K = k \times k' \times k''$, the compression ratio is given as D/K.

HOSVD, the FP, the FPT and the 3DDCT. Using these four methods, we compress the tensors to the sizes $32 \times 32 \times 32$, $16 \times 16 \times 16$ and $8 \times 8 \times 8$. The OU-ISIR dataset contains sequences of images of 34 people with nine different walking speeds. We use the sequences with walking speeds of 2, 4, 6, 8 and 10km/h for learning data and the sequences with walking speeds of 3, 5, 7 and 9km/h for test data. The recognition rate is defined as the successful label estimation ratio for 1000 label estimations. In each estimation of a label for a query, queries are randomly chosen from the test dataset. For the 1-, 2- and 3-modes, we evaluate the results for multilinear subspaces with sizes from one to the dimension of the compressed tensors.

Figure 4.9(a)-(c) shows the recognition rates for the four compression methods with three different sizes of the compressed tensors of OU-ISIR dataset. For the images of size $32 \times 32 \times 32$ shown in Fig. 4.9(a), the recognition rates for all four types of compressed tensor are almost coincident with those of the original tensors, if the compression ratio is higher than 10^3. If the compression ratio is less than 10^3, the recognition ratio of the FPT is higher than those of the HOSVD, the FP and the 3DDCT. The recognition ratio of the 3DDCT is lower than those of other methods since the silhouette images are binary images. For the images of sizes $16 \times 16 \times 16$ and $8 \times 8 \times 8$ shown in Figs. 4.9(b) and (c), although the recognition rates for the four types of compressed tensor are almost the same, the recognition rates are smaller than those for the original tensors. This recognition property depends on the size of the images, and the images used for the comparison are too small to evaluate our methods of the recognition of sequences of silhouette images. In Figs.4.9(a)-(c), the HOSVD and the FP give the same recognition rate. These results imply that the decomposition for the FP is independent of the number of iterations.

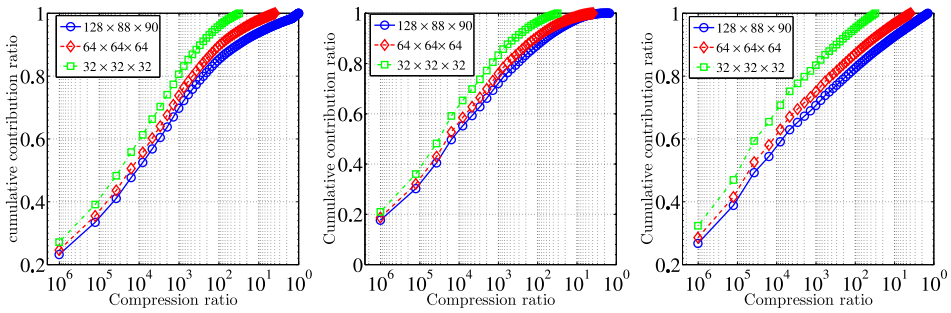
Figure 4.9(d)-(f) shows the recognition rates for the four compression methods with three different sizes of the compressed tensors. For the voxel images of sizes $32 \times 32 \times 32$ and $16 \times 16 \times 16$ shown in Figs. 4.9(d) and (e), the recognition rates for all four types of compressed tensor are almost coincident with those of the original tensors. For the voxel images of sizes $8 \times 8 \times 8$ shown in Fig. 4.9(f), the recognition rates for all four types of compressed tensor are almost coincident with those of the original tensors, if the compression ratio is higher than 10^4. Compared to the recognition rates of the sequences of silhouette images, the 3DDCT gives better approximation for ones of the volume data of livers since livers are essentially volumetric objects, which consists of textures of liver tissues.

Finally, using Intel Xeon X5570 Quad core 2.93GHz, we compare the computational times of the four reduction methods for OU-ISIR. In the comparison, for TPCA, we set 3 iterations for Algorithm 1. For computation of



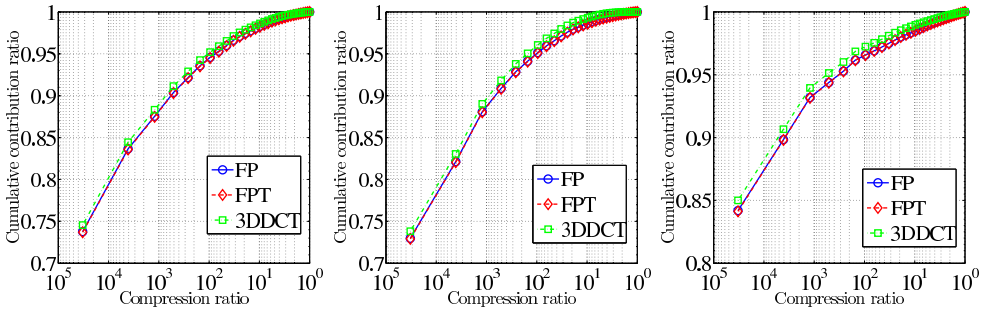
(a) $128 \times 88 \times 90$ (b) $64 \times 64 \times 64$ (c) $32 \times 32 \times 32$

Figure 4.6: Cumulative contribution ratio of three modes. (a)-(c) show the cumulative contribution ratios of the three modes for the input sizes $128 \times 88 \times 90$, $64 \times 64 \times 64$ and $34 \times 34 \times 34$ in Algorithm 1, respectively. The horizontal and vertical axes represent the compression ratio and cumulative contribution ratio, respectively. For the original size $D = 128 \times 88 \times 90$ and the reduced size $K = k \times k' \times k''$, the compression ratio is given as D/K.



(a) mode 1 (b) mode 2 (c) mode 3

Figure 4.7: Comparison of cumulative contribution ratio between full projection and full projection truncation. (a)-(c) show a comparison of the cumulative contribution ratio in the three modes. The horizontal and vertical axes represent the compression ratio and cumulative contribution ratio, respectively. For the original size $D = 128 \times 88 \times 90$ and the reduced size $K = k \times k' \times k''$, the compression ratio is given as D/K.



(a) mode 1 (b) mode 2 (c) mode 3

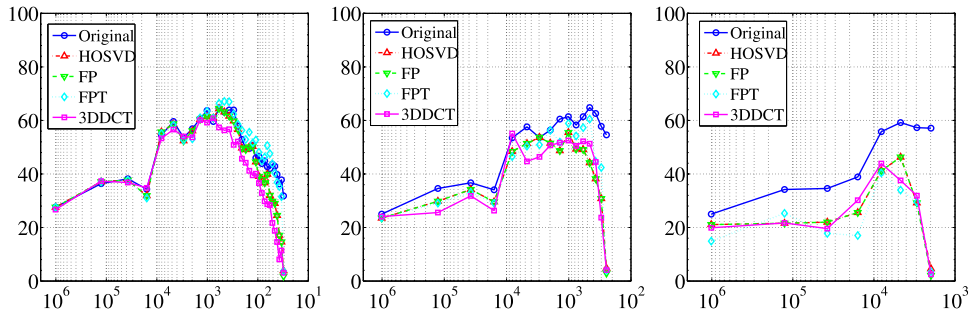
Figure 4.8: Comparison of cumulative contribution ratio for three types of compressed tensor. For the compression of tensors, we use Algorithm 1 and the 3DDCT. In Algorithm 1, we respectively adopt sizes of $128 \times 88 \times 90$ and $32 \times 32 \times 32$ for the computation by FP and FPT. For the three types of compressed tensor of $32 \times 32 \times 32$, we apply 10 iterations of Algorithm 1. In (a)-(c), the horizontal and vertical axes represent the compression ratio and cumulative contribution ratio, respectively. For the first reduced size $K_1 = 32 \times 32 \times 32$ and the second reduced size $K_2 = k \times k' \times k''$, the compression ratio is given as K_1/K_2.

3DDCT, we construct three DCT-II matrices. Therefore, we do not use FFT. Note that the computational times of TTP in the four methods are the same. Therefore, we show only the mean of the computational time. Figure. 4.10 shows the comparison of computational times for two datasets. The results show that the 3DDCT gives the fastest construction of projection matrices in the four methods.

For voxel images, the 3DDCT gives an acceptable approximation of the HOSVD, the FP and the FPT in the context of pattern recognition. Even for sequences of binary silhouette images, the 3DDCT gives an acceptable approximation of the HOSVD, the FP and the FPT in the context of pattern recognition. Moreover, from Figs. 4.7 and 4.8, and Figs.4.9(a)-(c), changes in the energies of the projected tensors and the CCRs of the eigenvalues in the decomposition of tensors in these methods are not important in the context of pattern recognition.

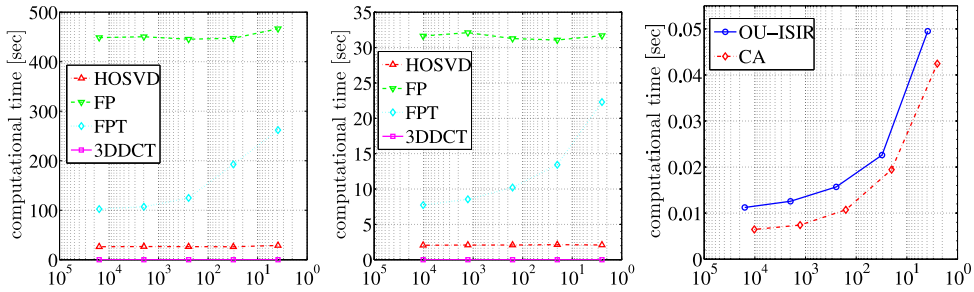
4.8.2 Volumetric Pattern

We present two examples for extraction of outline shapes of volume data, and abilities of our method for classification of volumetric data. For experiments,



(a) $32 \times 32 \times 32$ (b) $16 \times 16 \times 16$ (c) $8 \times 8 \times 8$

Figure 4.9: Recognition rates of gait patterns and liver data for original and compressed tensors. We adopt the reduces sizes of $32 \times 32 \times 32$, $16 \times 16 \times 16$ and $8 \times 8 \times 8$. (a)-(c) show the recognition rates for three reduced sizes of OU-ISIR. For compression, we use the HOSVD, FP, FPT and 3DDCT. In (a)-(c), the horizontal and vertical axes represent the compression ratio and recognition ratio [%], respectively. In (a)-(c) for the original reduced sizes $D = 128 \times 88 \times 90$, the compression ratio is given as D/K for reduced size k.



(a) (b) (c)

Figure 4.10: Computational time of dimension reduction for tensors of the order three. (a) and (b) show the computational time of construction of projection matrices for 306 sequences of silhouette images and 35 voxel images of livers, respectively . (c) shows the mean computational time of projecting images to low-dimensional tensor space for OU-ISIR and CA datasets. In (a) and (b), we compare the HOSVD, FP, FPT and 3DDCT. In (a)-(c), the vertical and horizontal axes represent the computational time and compression ratio, respectively.

Table 4.3: Sizes and number of volumetric data of livers. ♯data represents the number of livers obtained from different patients. The data size is the original size of the volumetric data. The reduced data size is the size of the volume data after tensor-based reduction.

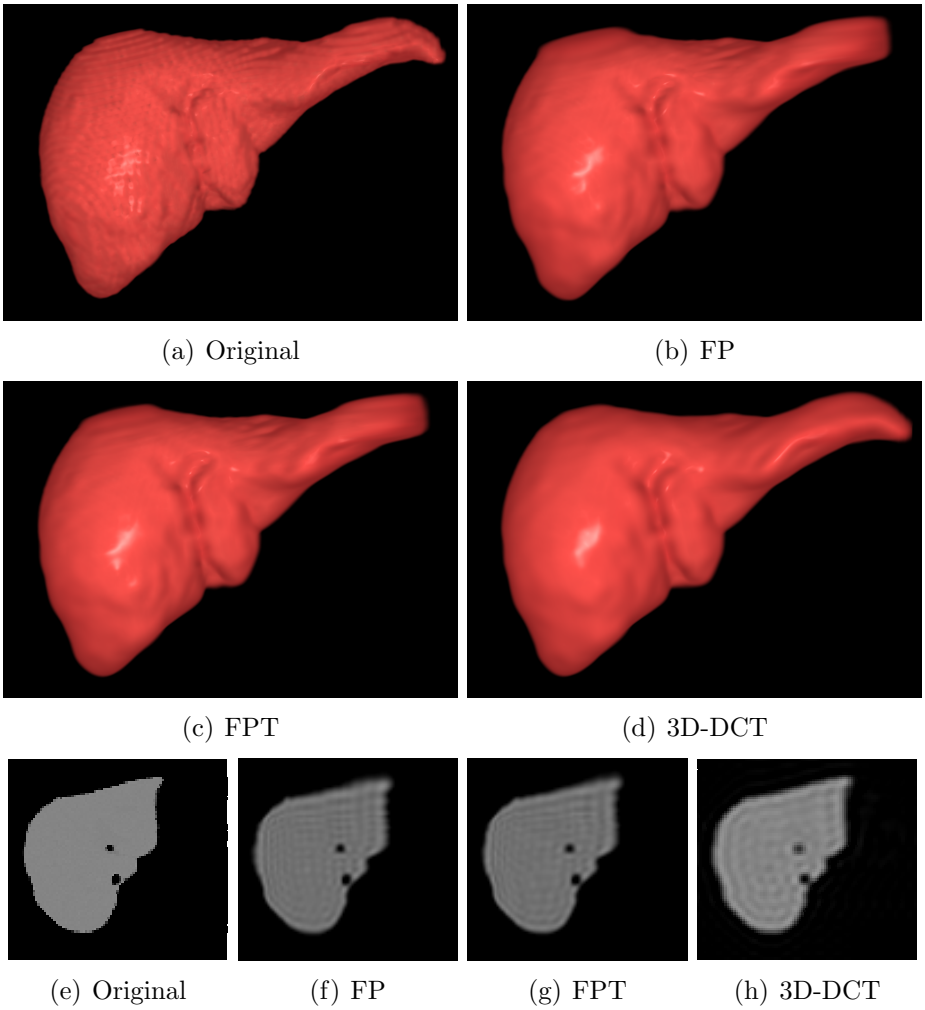
	♯data	data size [voxel]	reduced data size [voxel]
Volumetric liver data	32	$89 \times 97 \times 76$	$32 \times 32 \times 32$

we use the voxel images of human livers obtained as CT images. This image set contains 25 male-livers and seven female-livers. Note that these voxel images are aligned to their centre of gravity. In the experiments, we project these voxel images to small size tensors. For the projections, we adopt TPCA and the 3D-DCT. In the iterative method of TPCA, setting the number of bases to the size of the original tensors in Algorithm 1, we call the method full projection (FP). If we set the number of bases to smaller than the size of the original tensors in Algorithm 1, we call the method full projection truncation (FPT). Table 1 summarises the sizes and numbers of original and dimension-reduced voxel images.

Firstly, we show the approximation of a voxel image of a liver by three methods. The FP, FPT and 3D-DCT reduce the size of the data from $89 \times 97 \times 76$ voxels to $32 \times 32 \times 32$ voxels. Figure 4.11 illustrates volume rendering of original data and reconstructed data by these compressed tensors. Compared to Figs. 4.11 (a) and 4.11 (e), in Figs. 4.11 (b)-(c) and (f)-(h), the FP, FPT and 3D-DCT preserve outline shapes of liver. In Figs. 4.11, the reconstructed data by the 3D-DCT gives a closer outline shape and more similar interior texture to those of the original than the FP and FPT. In Figs. 4.11, these results show that projections to small-size tensors extract outline shapes.

For the analysis of projected data by the FP, FPT and 3D-DCT, we decompose these projected tensors by Algorithm 1. Here, we set the size of bases in Algorithm 1 to $32 \times 32 \times 32$ and use 35 projected tensors of livers for each reduction methods. In decompositions, we reordered eigenvalues $\lambda_i^{(j)}$, $j = 1, 2, 3$, $i = 1, 2, \ldots, 32$ of the three modes to λ_i, $i = 1, 2, \ldots, 96$ in the descending order. Figure 4.12 shows the cumulative contribution ratios of reordered eigenvalues for the projected tensors obtained by the FP, FPT and 3D-DCT. Figure 4.13 illustrates reconstructed data obtained by using the 20 major principal components.

In Fig. 4.12, profiles of curves for three methods are almost coincident while the CCR of the 3D-DCT is a little bit higher than the others. In three methods, the CCRs become higher than 0.8 if we select more than 19 major



(a) Original (b) FP

(c) FPT (d) 3D-DCT

(e) Original (f) FP (g) FPT (h) 3D-DCT

Figure 4.11: Original and reconstructed volumetric data of liver data. (a) shows the rendering of original data. (b)-(d) show the rendering of reconstructed data after the FP, FPT and 3DDCT, respectively. (e)-(f) illustrate axial slice images of these volumetric data in (a)-(d), respectively. The sizes of reduced tensors are shown in Table. 1.

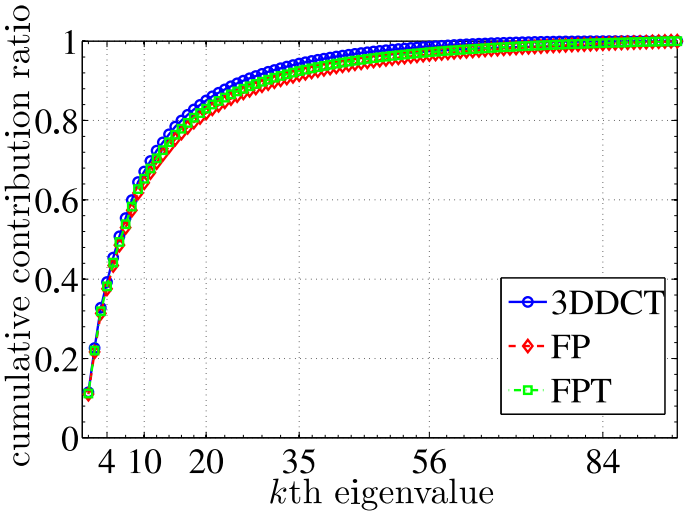
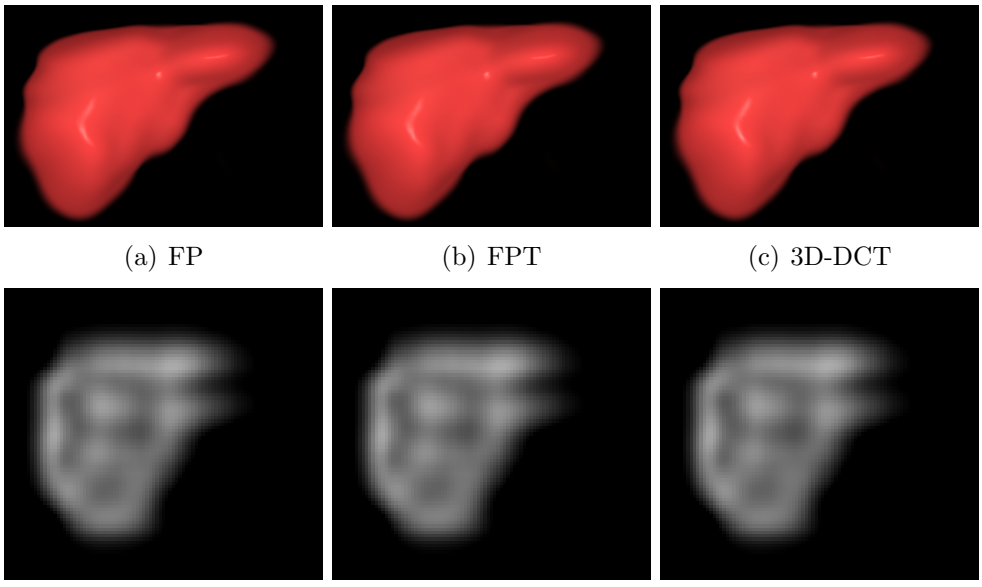


Figure 4.12: Cumulative contribution ratios for three compressed tensors. For compression, we adopt FP, FPT and 3D-DCT. For the computation of the cumulative contribution ratio of eigenvalues obtained by the FP, we used all eigenvalues of modes 1, 2 and 3 after sorting them into descending order.

principal components. In Fig. 4.13, shapes and interior texture for three methods are almost the same. In Figs. 4.13(d)-(f), the interior texture of a liver is not preserved and the outer shape is burred. In these results for three methods, major principal components represent outline shapes.

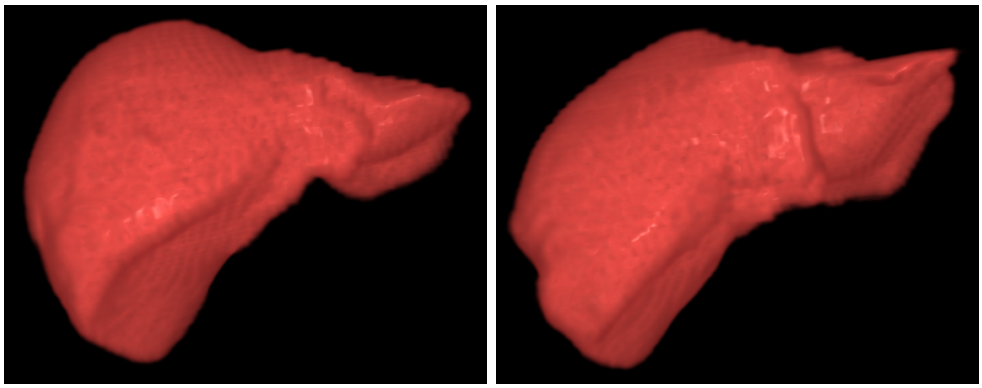
Secondly, we show results of the classification of voxel images of livers by the TSM. For the classification, we use 25 male-livers and seven female-livers since the sizes and shapes of livers between male and female are statistically different. Figures 4.14(a) and 4.14(b) illustrates the examples of livers of male and female, respectively. We use the voxel images of livers of 13 males and 4 females as training data. The residual voxel images are used as test data. In the recognition, we estimate the gender of livers. The recognition rate is defined as the successful estimation ratio for 1000 gender estimations. In each estimation of a gender for a query, queries are randomly chosen from the test dataset. For the 1-, 2- and 3-modes, we evaluate the results for multilinear subspaces with sizes from one to the dimension of the rejected tensors. Figure 4.15 shows the results of the classification. The TSM give 90 % recognition rate at the best with tensor subspace spanned by every two major principal axis of the three modes.



(a) FP (b) FPT (c) 3D-DCT

(d) FP (e) FPT (f) 3D-DCT

Figure 4.13: Reconstruction by using only major principal components of the decomposition by the FP. Top and bottom rows illustrate volume rendering and axial slice of reconstructed data, respectively. For reconstruction, we use the 20 major principal components. Left, middle and right columns illustrate the results for the tensors projected by the FP, FPT and 3D-DCT.



(a) Male (b) Female

Figure 4.14: (a) and (b) illustrate the examples of livers of male and female, respectively.

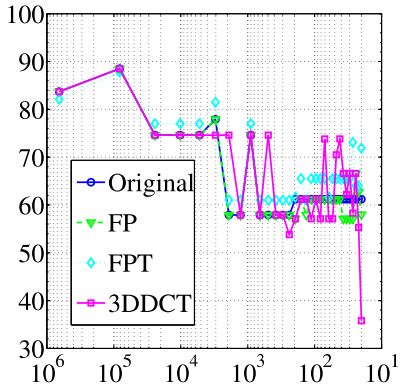


Figure 4.15: Recognition rates of liver data for original and compressed tensors. For compression, we use the HOSVD, FP, FPT and 3D-DCT. The horizontal and vertical axes represent the compression ratio and recognition ratio [%], respectively. For the original size $D = 89 \times 97 \times 76$ and the reduced size $K = k \times k' \times k''$, the compression ratio is given as D/K for reduced size k.

4.8.3 Spatio-Temporal Pattern

We comparatively evaluate performance of tensor subspace method and mutual tensor subspace method in classification of volumetric data. For this evaluation, we adopt the following steps.

1. Extract the volumetric data of the left ventricles from cardiac MRI dataset.
2. Reduce the dimension of the extracted data by using four methods.
3. Divide the dimension-reduced data to taring and test data.
4. Construct tensor subspace of each categories by applying the the TPCA to the training data.
5. Classify a query (or query subspace) of a category in test data by using tensor-based classifiers.

We extract sequences of volumetric data of left ventricle from cardiac MRI dataset [10], since we need to validate classification of normal organs before abnormality detection. For the extraction, we use the landmarks of endocardium of left ventricle. These landmarks are manually given and provided as the part of the dataset. Table 4.4 summarises the number and the

Table 4.4: Sizes and number of volumetric data of left ventricles. ♯category represents the number of individuals. ♯data/category represents the number of frames in one sequence of left ventricles. The data size is the original size of the volumetric data. The reduced data size is the size of the volumetric data after reduction. We set $d \in \{8, 16, 32\}$.

	♯category	♯data /category	data size [voxel]	reduced data size [voxel]
Volumetric data	17	20	$81 \times 81 \times 63$	$d \times d \times d$

size of the extracted volumetric data at all phases. Figure 4.16 illustrates the extracted sequences of volumetric data for 17 patients. Since a beating heart deforms its volumetric shape, we obtain third-order tensors representing shape of heart with geometrical perturbation from a sequence.

For the dimension reduction of volumetric data, we use the TPCA and 3D-DCT. For the practical computation of the TPCA, we use the HOSVD and MPCA. In the MPCA, if we set the number of bases to the size of the original tensors in Algorithm 1, we call the method full projection (FP). If we set the number of bases to fewer than the size of the original tensors in Algorithm 1, we call the method the full-projection truncation (FPT). For the dimension reduction by the HOSVD, FP and FPT, we apply these methods to all the extracted volumetric data in all categories. For the evaluation the robustness and stabilizes of methods with respect to the sizes of the data, we set the sizes of the dimension-reduced data to $8 \times 8 \times 8$, $16 \times 16 \times 16$ and $32 \times 32 \times 32$.

Figure 4.17 illustrates the comparison between original and dimension-reduced data by the three-methods. In Fig. 4.17(a)-(d), volume rendering of the original and reconstructed volume data are presented. For the data reduced by the FP and FPT, the shapes of volumetric data reconstructed from the compressed data are almost the same in their appearances. The reconstructed data from the data reduced by the 3DDCT is the closest shape to the shape of original volumetric data. In Figs 4.17(e)-(h), the differences of appearances between the sagittal slices of reconstructed data and original shape are compared. Compared to the original data shown in Fig 4.17(a), the 3D-DCT gives blurred inner texture as shown in Fig 4.17(h). As shown in Figs 4.17(f) and (g), the dimension reduction by the FP and FPT extract outline shapes of ventricle without inner texture. Figure 4.18 illustrates reconstructed data from principal components of dimension-reduced volume data. This result show that the principal components of the dimension-

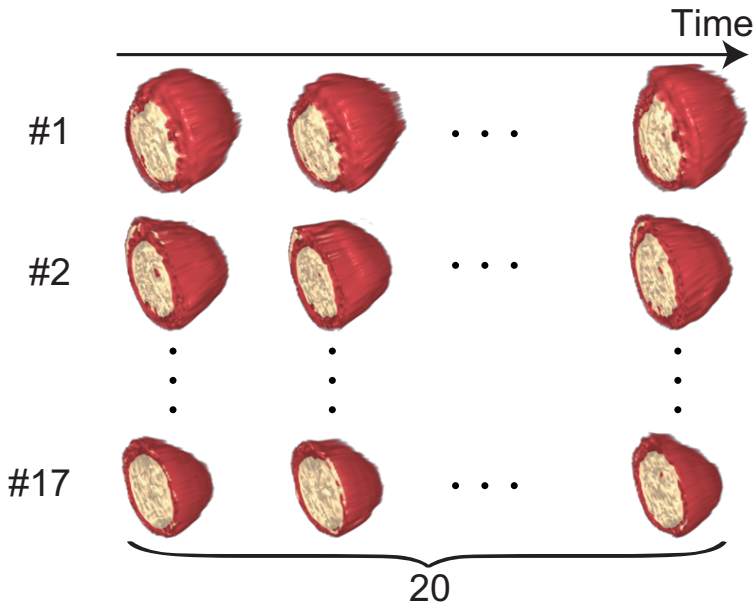
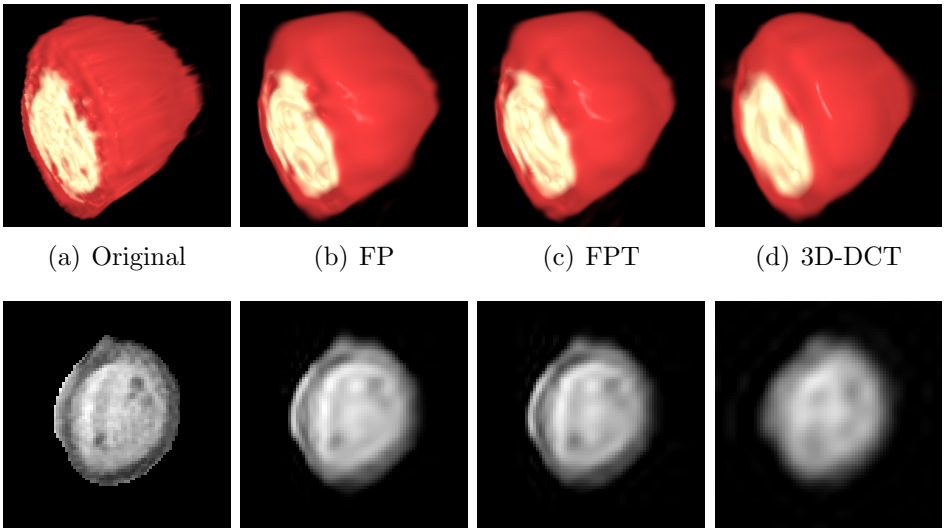


Figure 4.16: Illustration of extracted cardiac MRI dataset. These sequences of volumetric data are extracted from cardiac MRI dataset with landmarks of endocardium of left ventricles [10]. As shown in Table 1, we have 17 sequences of volumetric data of left ventricle for 17 patients. Each sequence of volumetric data represents one cardiac beat by 20 frames. Every sequence starts with maximally expanded state. Red and white parts of volume rendering of the data represent muscle and inner space of left ventricles. We set the center of the first sagittal slice of each volume data to the center of the slice.

reduced volume data are almost the same.

In the dimension-reduced data, each sequences consist from 20 frames. We use odd and even frames in dimension reduced data as training and test data, respectively. Applying the FP to training data of each category, we construct tensor subspace of 17 categories for the TSM and MTSM. The TSM and MTSM are robust classification methods against geometrical perturbations. Therefore, we use only odd frame for construction of tensor subspaces of categories to evaluate these robustness. If the TSM and MTSM classify a category of even frame, we conclude that these classifiers are robust to the small geometrical change between frames.

The recognition rate is defined as the successful classification ration of individuals in 1000 classifications. We use the TSM and MTSM as classifiers. In the selection of a query for the TSM, we randomly select one of 17



(a) Original (b) FP (c) FPT (d) 3D-DCT

(e) Original (f) FP (g) FPT (h) 3D-DCT

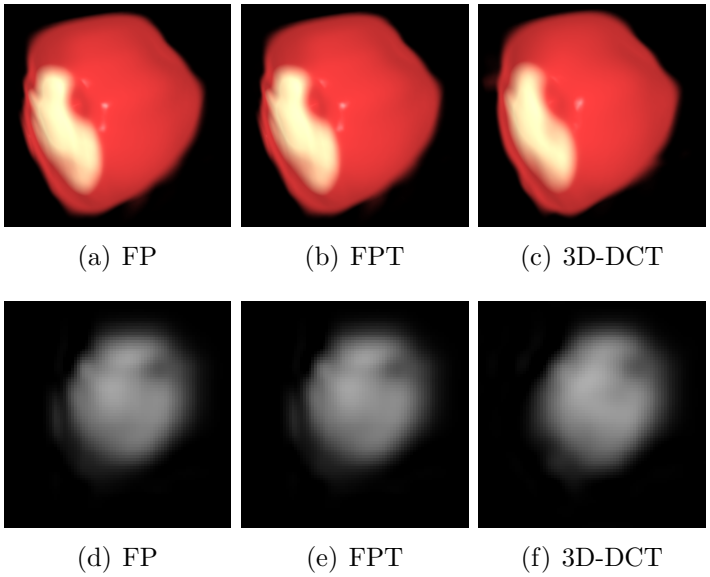
Figure 4.17: Shape and inner texture of reconstructed volume data of left ventricle from compressed data. Upper and lower rows show volume rendering and sagittal slice of the volumetric data, respectively. In (a)-(d), red and white parts depict the muscle of heart and inner of heart, respectively, for original and approximation by the FP, the FPT and the 3D-DCT. In these approximation, the data are reduced to the size $16 \times 16 \times 16$.

individuals and randomly select one of test data of the individual. From a collection of input data of the heart sequence of a patient, we constructed the subspace of queries for the MTSM. After randomly select an individual, from one to three queries are randomly selected from test data of the selected individual. Applying the FP to the selected queries, we construct a query tensor subspace for the MTSM. Figures 4.19 and 4.20 show recognition rate for left ventricles by the TSM and the MTSM, respectively.

In Fig 4.19, the profiles of recognition curves for the HOSVD, FP, FPT and 3D-DCT are almost the same in the higher compression ratio than 10^3.

Table 4.5: Reconstruction error of volumetric data. The reconstruction error is given by distance between tensors of the original and reconstructed volumetric data.

	FP	FPT	3D-DCT
Reconstruction Error	25.9×10^3	25.9×10^3	8.14×10^3



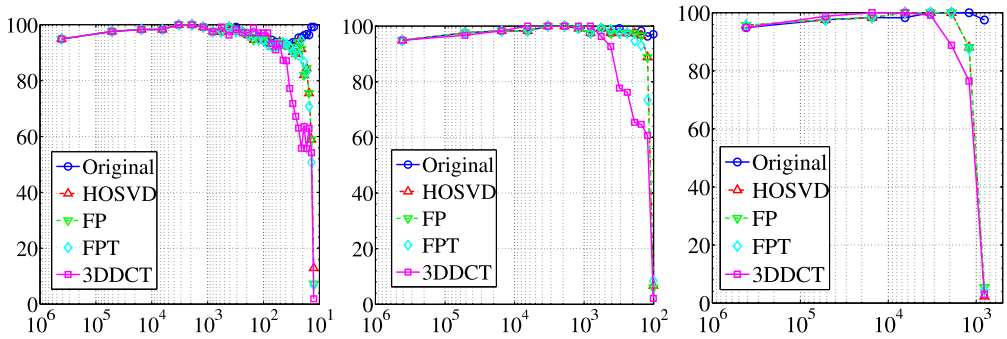
(a) FP (b) FPT (c) 3D-DCT

(d) FP (e) FPT (f) 3D-DCT

Figure 4.18: Extracted principal components of dimension-reduced volume data. For the data dimension reduced by the FP, FPT and 3D-DCT, we apply the FP. Using the extracted principal component, we reconstruct volumetric data. For the extraction, we select the 20 principal eigenvectors of ones of three modes.

Furthermore, for the higher compression ratio than 10^3, the dimension reduced data by four methods derive almost same recognition rates. These recognition rates are the same recognition rate of the tensors of original size. Moreover, the TSM with major five eigenvenctors in each modes processes accurate recognition. Figure 4.20 (a)-(c) show that the MTSM for the query subspace spanned by one query. The results show that the recognition properties are almost the same for data with $8 \times 8 \times 8$, $16 \times 16 \times 16$ and $32 \times 32 \times 32$. The results in Figs 4.20 (d)-(i) show that the MSM achive more robust recognition against small geometrical perturbations by using a query subspace than the TSM, since a query subspace spanned by a few queries with geometrical perturbations.

These numerical examples show that the 3DDCT accurately approximates performance of the TPCA. Furthermore, recognition of three-way data by the TSM and MTSM is accurate and robust for volumetric data contain geometrical perturbations as temporal deformation. Moreover, the mutual tensor subspace method achive more robust recognition than the tensor subspace method, since the mutual tensor subspace method based on both category tensor subspace and query tensor subspace.



(a) $32 \times 32 \times 32$ (b) $16 \times 16 \times 16$ (c) $8 \times 8 \times 8$

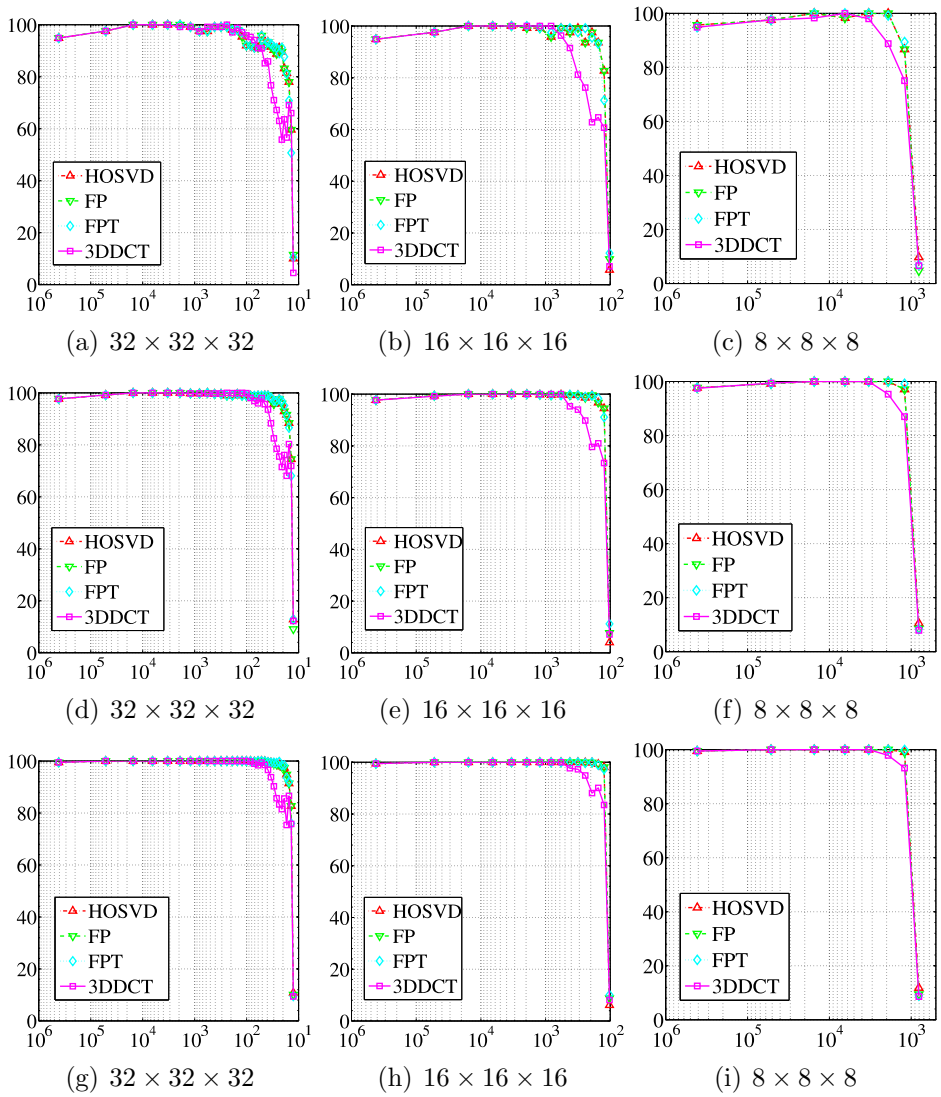
Figure 4.19: Recognition rates of the left ventricles for original and compressed tensors. We use tensor subspace method as classifier. The data are reduced to $32 \times 32 \times 32$, $16 \times 16 \times 16$ and $8 \times 8 \times 8$. The HOSVD, FP, FPT and 3D-DCT are used for the reduction. Vertical and horizontal axes represent recognition rate and compression ratio, respectively. For the original size $D = 81 \times 81 \times 63$ and reduced size $K = k \times k' \times k'$, the compression ratio is given by D/K.

4.8.4 Geometry of Multilinear Subspace

Gait Sequence

Using the Wasserstein distance for tensor subspaces, we analyse changes of multilinear structure of gait patterns. For this analysis, we use silhouette images in OU-ISIR treadmill A dataset shown in Fig. 4.21. The OU-ISIR dataset includes several sequences of different walking person in different walking speeds. We pick up a sequence of a person of 4 steps in waking speed of 2km/h for computation. This sequence consists of 90 frames of 128×88 pixels.

We first compute the Wasserstein distance between the first and ith frames using eigenvectors of mode-1 and -2 obtained by the singular value decomposition and compare the Wasserstein distance with Euclidean distance. Figure 4.22 shows the Wasserstein distances compared with Euclidean distance between the first and ith frames. Wasserstein distance more clearly quantifies the difference between images than Euclidean distance, while the shape of curves of Wasserstein and Euclidean distances are similar. Compared with Wasserstein distances for mode-1, Wasserstein distances for mode-2 more clearly represent difference between images, since the shape of the distance for mode-2 is similar to the shape of the Wasserstein distance between images.



(a) $32 \times 32 \times 32$ (b) $16 \times 16 \times 16$ (c) $8 \times 8 \times 8$

(d) $32 \times 32 \times 32$ (e) $16 \times 16 \times 16$ (f) $8 \times 8 \times 8$

(g) $32 \times 32 \times 32$ (h) $16 \times 16 \times 16$ (i) $8 \times 8 \times 8$

Figure 4.20: Recognition rates of left ventricles for compressed tensors. We adopt the reduces sizes of $32 \times 32 \times 32$, $16 \times 16 \times 16$ and $8 \times 8 \times 8$. For compression, we use the HOSVD, FP, FPT and 3D-DCT. In the mutual tensor subspace method, input is a query subspace. The query subspace is spanned by a few queries. To construct a query subspace, we use one, two and three queries. Top, middle and bottom row show recognition rates for the case of one, two and three queries, respectively. Vertical and horizontal axes represent recognition rate and compression ratio, respectively. For the original size $D = 81 \times 81 \times 63$ and reduces size $K = k \times k' \times k'$, the compression ratio is given by D/K.

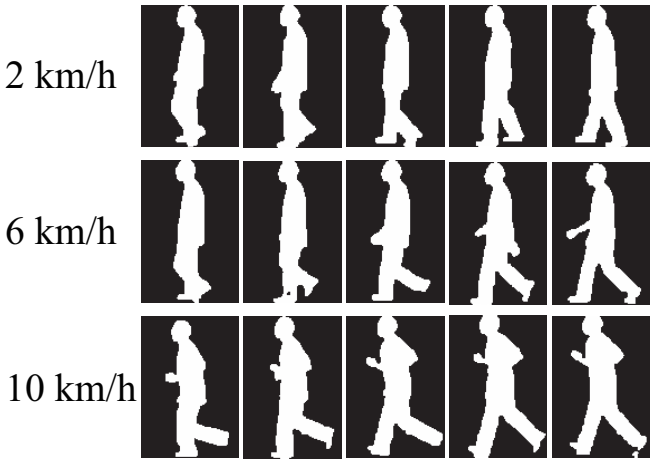


Figure 4.21: Silhouette images of a walking person in OU-ISIR dataset. The top, middle and bottom row represent sequence of four steps in different speeds for a man.

We second analyse what major principal components express. Figure 4.23 illustrates contribution ratios and cumulative contribution ratios of eigenvalues for the first frame. In Fig. 4.23, about 70 % of cumulative contribution ratio concentrates on major six eigenvalues. Using major principal components to coincident to these six eigenvalues, that is, from the first to six eigenvectors for mode-1 and -2, we reconstruct the first frame as shown in Figs. 4.24 (a)-(c). Using minor principal components, that is, the principal components but major principal components, we reconstruct the first frame as shown in Figs. 4.24 (d)-(f). In Fig. 4.24 (a), reconstructed image represents mean of distributions of an unfolded tensor for each mode. Figures 4.24 (b), (d) and (e) show that additional second and third eigenvectors for both modes represent change along vertical and horizontal direction. Furthermore, Figs 4.24 (c), (e) and (f) show that additional forth, fifth and sixth eigenvectors for both modes represent second-order derivative directions.

Third, we examine relation among eigenvectors of two images. We use successive two frames in a sequence shown in Fig. 4.25. By using singular value decomposition, we compute eigenvectors of mode-1 and -2 for the two frames. For these eigenvectors, we compute inner products among them. Figure 4.26 summarises the inner products. In Figs. 4.26 (a) and (b), from the first to fifth eigenvectors, the eigenvectors of the same order for two frames are almost coincident. Furthermore, we compute the difference between reconstructed images. Figure 4.27 shows the differences between the images reconstructed from selected eigenvectors. Compared to the Fig. 4.27,

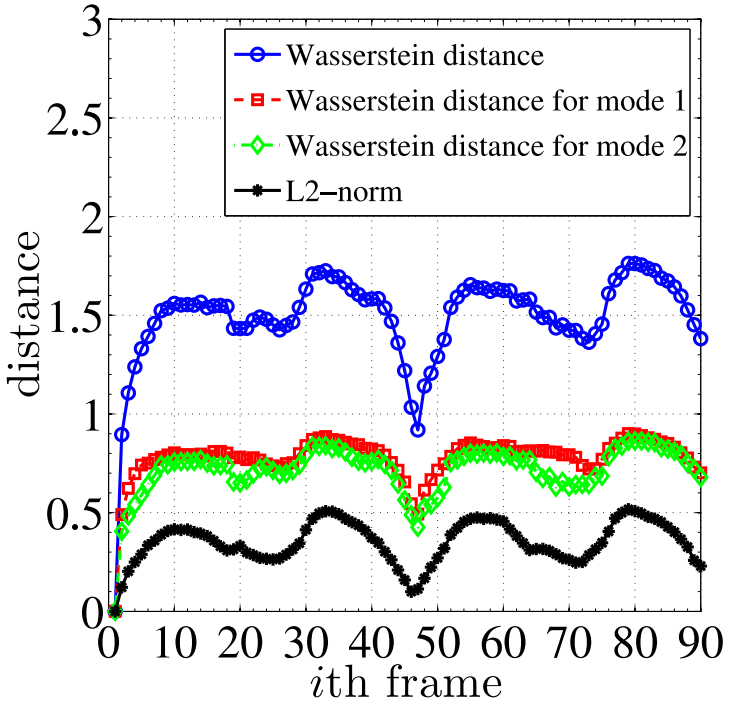
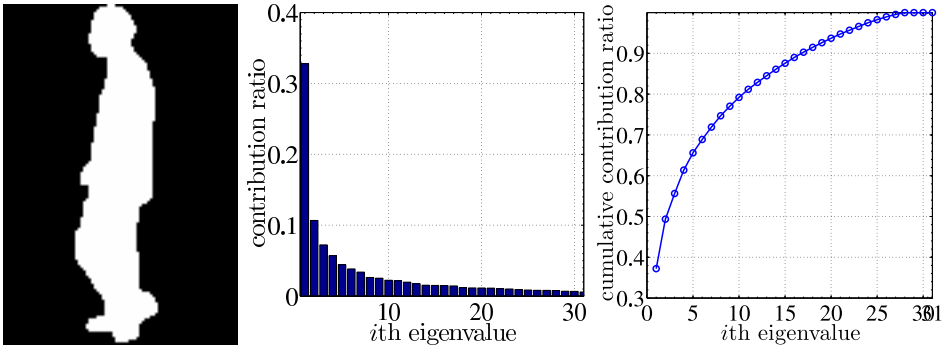


Figure 4.22: Wasserstein distances and Euclidean distance between first and ith frames for $i = 1, 2, \ldots, 90$. Plotted Euclidean distance is the relative distance for the L_2-norm of the first frame. This relative distance is defined by $\|\boldsymbol{X}_1 - \boldsymbol{X}_i\|_{\mathrm{F}}/\|\boldsymbol{X}_1\|_{\mathrm{F}}$, where $\boldsymbol{X}_1$ and $\boldsymbol{X}_i$ are the first and ith frames, and $\|\cdot\|_{\mathrm{F}}$ is Frobenius norm.



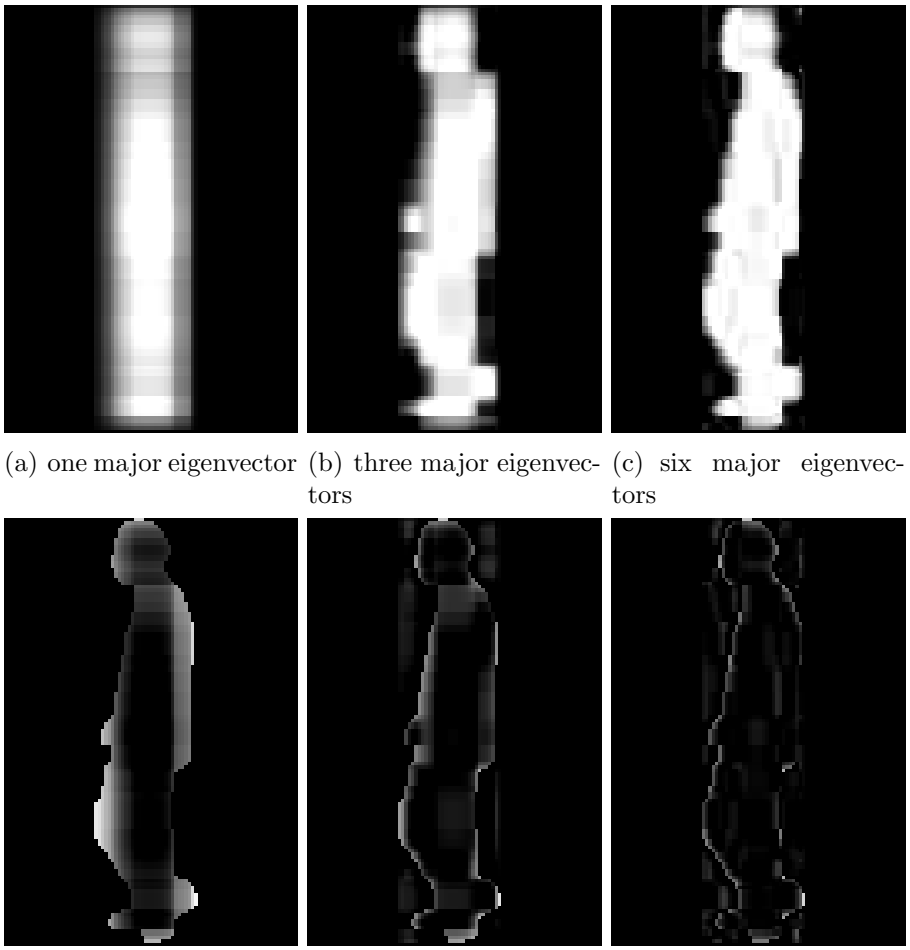
(a) (b) (c)

Figure 4.23: Example of decomposition of an image. (a) The first frame of walking person in 2km/h of OU-ISIR treadmill dataset. (b) contribution ratio of eigenvalues obtained by singular value decomposition. (c) cumulative contribution ratio of eigenvalues obtained by singular value decomposition

in Figs. 4.27, the differences between frames reconstructed from one, three and six major eigenvectors are not correct. On the other hand, the difference between frames reconstructed from minor eigenvectors are close to the difference between original successive frames. These results imply that analysis of minor components is important for analysis of time sequence data.

Forth, we compute Wasserstein distance between the first and ith frames by using major and minor eigenvectors. Figure 4.28 summarises the results of the computation of the Wasserstein distance. Figure 4.29 summarise the results of the computation of the Wasserstein distance for each mode. In Fig. 4.28, the difference between two frames are mainly depends on the second, third, forth, fifth and sixth eigenvectors. These eigenvectors represents change of successive frames. Using these eigenvectors, we can reconstruct the part around boundary of a silhouette image as shown in Fig 4.24. Furthermore, in Fig. 4.29, the Wasserstein distance for mode-2 is larger than the one for mode-1. This result imply that the change between the successive frames are represent by eigenvectors for mode-2.

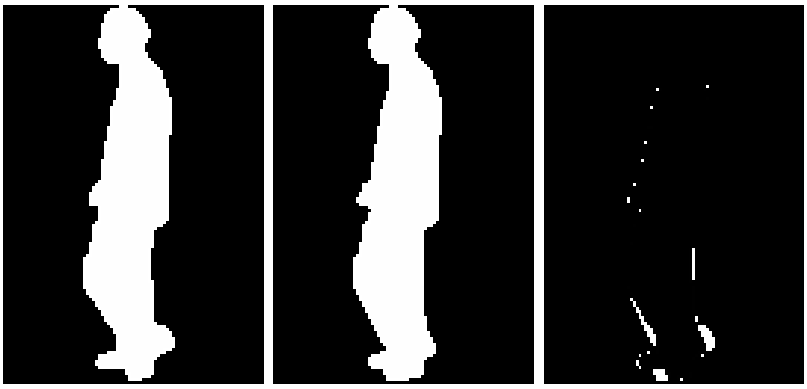
Fifth, we compute the Wasserstein distance between subspaces of different categories. Using tensor principal component analysis for second-order tensors, we compute tensor subspaces of walking sequences for different persons. Using these tensor subspaces, we compute the Wasserstein distances between subspaces of the first and kth categories. Figure 4.30 summarises these Wasserstein distances for all modes and each mode. In many the distances for the categories, the distance for mode-1 is larger than the one for



(a) one major eigenvector (b) three major eigenvectors (c) six major eigenvectors

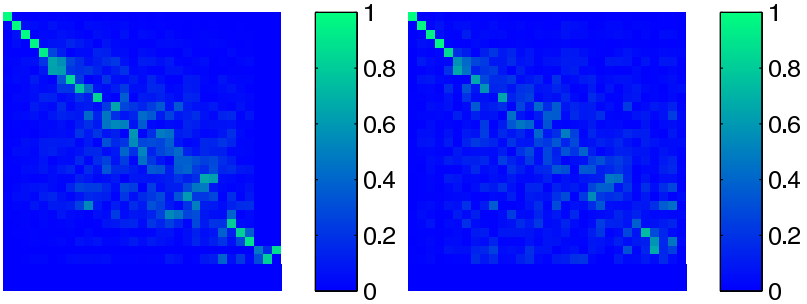
(d) 30 minor eigenvectors (e) 28 minor eigenvectors (f) 25 minor eigenvectors

Figure 4.24: Reconstruction of the first frame. In (a), (b) and (c), the first frame reconstructed by using one, three, six major eigenvectors for mode-1 and -2. In (d), (e) and (f),the first frame reconstructed by using 30, 28 and 25 minor eigenvectors for mode-1 and -2.



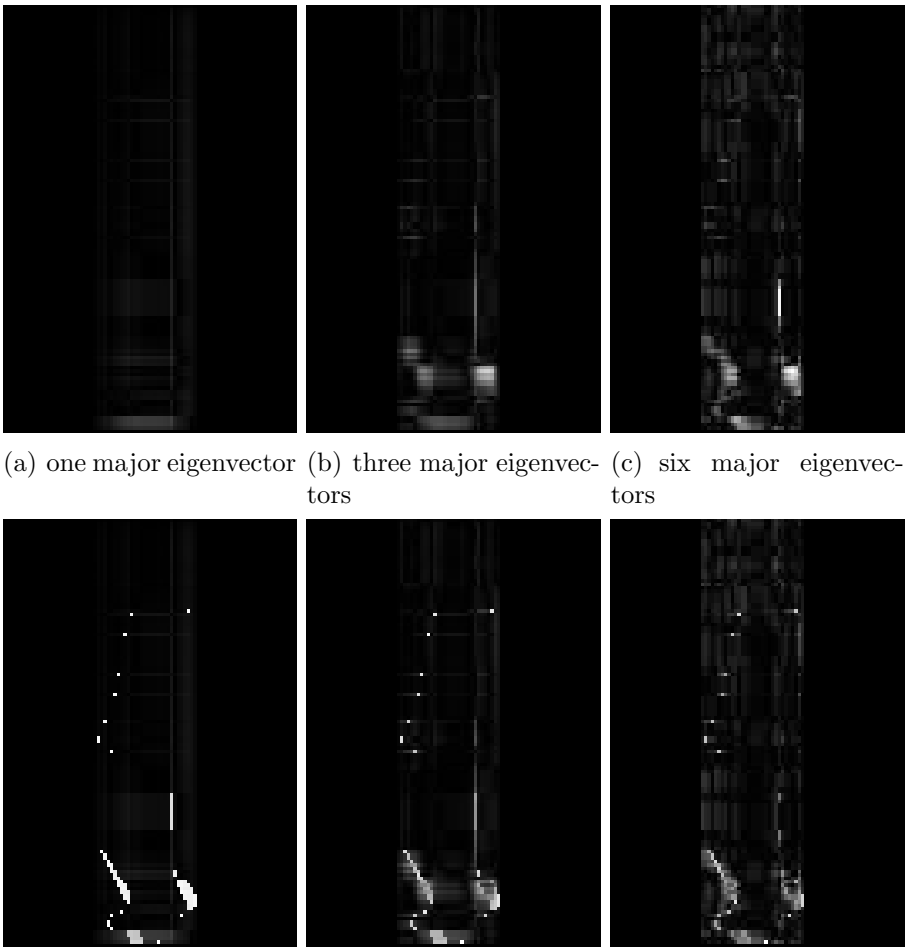
(a) (b) (c)

Figure 4.25: Successive two frames in a sequence. (a) Pre frame. (b) Post frame. (c) The difference of two frames. For visualisation, each pixel of the difference is displayed in its absolute value.



(a) mode 1 (b) mode 2

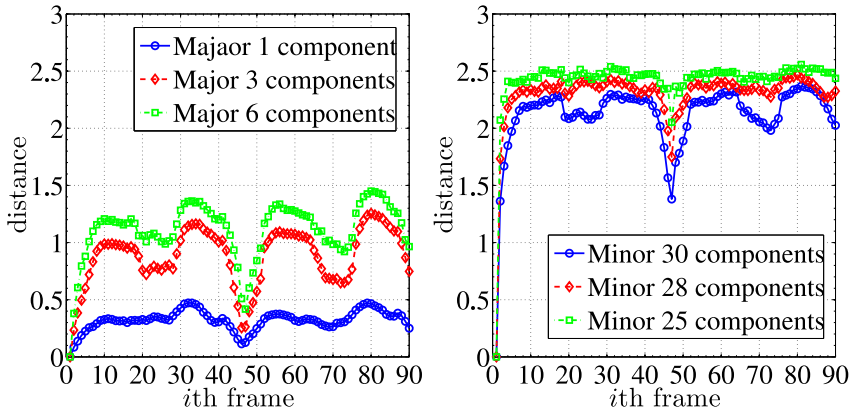
Figure 4.26: Absolute value of inner products for eigenvectors between 1st and 2nd frames. (a) and (b) show the inner products for eigenvectors of mode 1 and 2, respectively. In (a) and (b), from top to bottom, rows represent the eigenvectors of the second frame in descending order of eigenvalues. In (a) and (b), from left to right, columns represent the eigenvectors of the first frame in descending order of eigenvalues.



(a) one major eigenvector (b) three major eigenvectors (c) six major eigenvectors

(d) 30 minor eigenvectors (e) 28 minor eigenvectors (f) 25 minor eigenvectors

Figure 4.27: Difference between reconstructed images.



(a) Major principal components (b) Minor principal components

Figure 4.28: Wasserstein distances between first and ith frames for major and minor principal components.

the other modes.

Sixth, we reconstruct a sequence of a person displayed in Fig. 4.23 from major principal components of the third-order principal component analysis. Figures 4.31 (a), (b) and (c) show the first frames of the sequence reconstructed by using one, three and six major principal components of each mode. In the third-order tensor, changes among the frames in a sequence are expressed by bases for mode-3 while changes among the frames are expressed by bases for mode-1 and -2 in the second-order principal component analysis the. Compared with the reconstructed frame in Fig. 4.24, the frames of the reconstructed sequences are blurred.

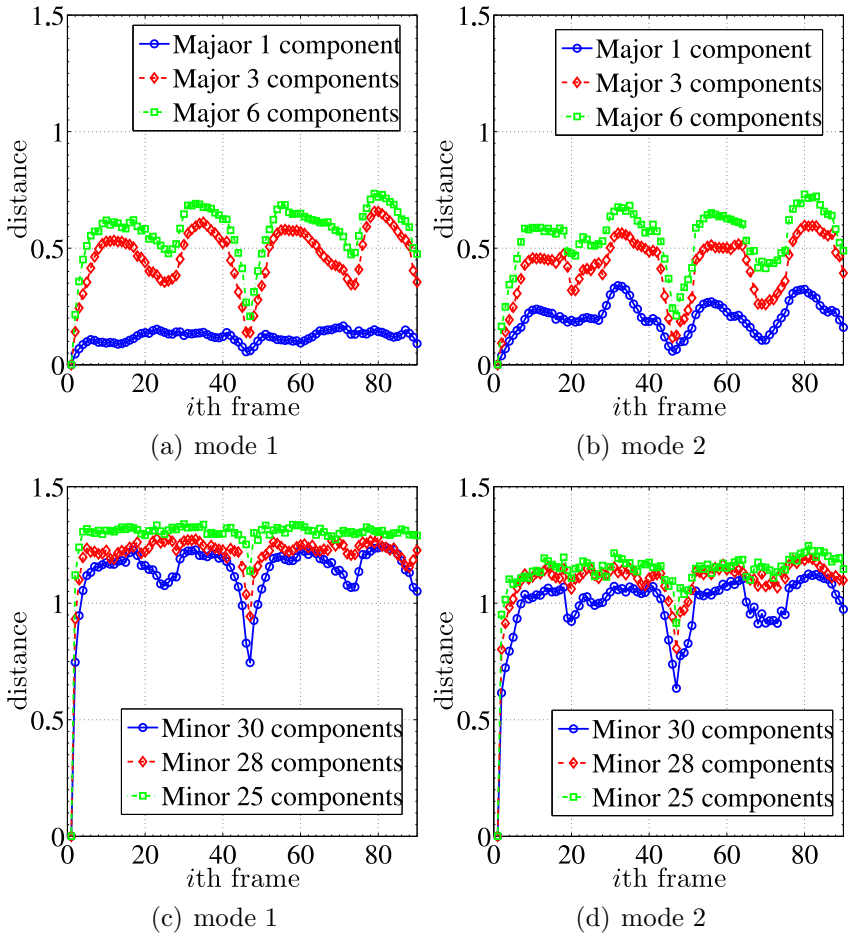
Seventh, we compute the Wasserstein distances among the subspace of categories by using major and minor principal components, respectively. Figures 4.32 (a), (c) and (e) show the distances computed by using 1, 3 and 6 major eigenvectors for each mode, respectively. Figures 4.32 (b), (d) and (f) show the distances computed by using minor eigenvectors except 1, 3 and 6 major eigenvectors for each mode, respectively. In Fig. 4.32, the distances for mode-3 for all combinations of two subspaces are almost the same. This result implies that the differences among categories in mode-3 are expressed by only major 6 eigenvectors of mode 3. Furthermore, in the Fig. 4.32 (f) the variance of distances of mode-1 and -2 are smaller than the ones in Figs. 4.32 (a), (b), (c), (d) and (e). This results imply that the differences among categories in mode-1 and -2 also expressed by only major 6 eigenvectors of mode-1 and -2.

Cardiac Sequences

As the second example, using the Wasserstein distance for tensor subspace, we analyse the changes of multilinear structure of beating human heart. For the analysis, we use sequences of slice images of left ventricle in cardiac MRI dataset. The cardiac MRI dataset includes 17 sequences of volumetric images of human body captured by MRI. The dataset also landmarks to specify the area of a left ventricle for each frame of sequences. Using the landmarks, we extract sequences of volumetric data of $81 \times 81 \times 63$ voxels. From these sequences, we extract eighth slices of each volumetric data and obtain sequences of slices. We use these extract sequences of 20 slices of 81×81 pixels. These sequences express each one cycle of beating of left ventricles. Figure 4.33 shows an example of extracted sequence of eight slices. In Fig. 4.33, heart shrinks and expands from (a) to (t).

We first compute the Wasserstein distance between the first and ith frames of a sequence using eigenvectors of mode-1 and -2 obtained by the singular value decomposition and compare it with Euclidean distance. Figure 4.34 shows the Wasserstein distances compared with Euclidean distance between the first and ith frames. Wasserstein distance more clearly quantifies the difference between images than Euclidean distance, while the shape of curves of Wasserstein and Euclidean distances are similar. Compared with Wasserstein distances for mode-1, Wasserstein distances for mode-2 more clearly represent difference between images, since the shape of the distance for mode-2 is similar to the shape of the Wasserstein distance between images.

We second analyse what major principal components express. Figure 4.35 illustrates contribution ratios and cumulative contribution ratios of eigenvalues for the first frame. In Fig. 4.35, about 66 % of cumulative contribution ratio concentrates on major six eigenvalues. Using major principal components to coincident to these six eigenvalues, that is, from the first to six eigenvectors for mode-1 and -2, we reconstruct the first frame as shown in Figs. 4.36 (a)-(c). Using minor principal components, that is, the principal components but major principal components, we reconstruct the first frame as shown in Figs. 4.36 (d)-(f). In Fig. 4.36 (a), reconstructed image represents mean of distributions of an unfolded tensor for each mode. Figures 4.36 (b), (d) and (e) show that additional second and third eigenvectors for both modes represent change along vertical and horizontal direction. Furthermore, Figs 4.36 (c), (e) and (f) show that additional forth, fifth and sixth eigenvectors for both modes represent second-order derivative directions.



(a) mode 1 (b) mode 2

(c) mode 1 (d) mode 2

Figure 4.29: Wasserstein distances for mode-1 and -2. Top and bottom rows summarise the Wasserstein distances by using major and minor principal components, respectively.

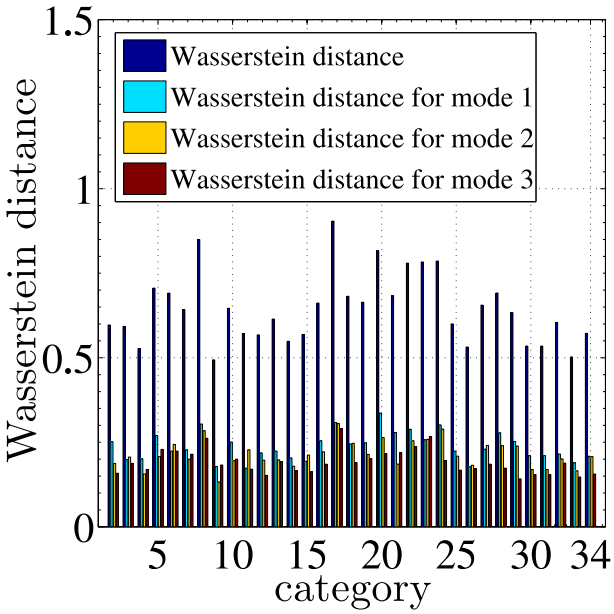
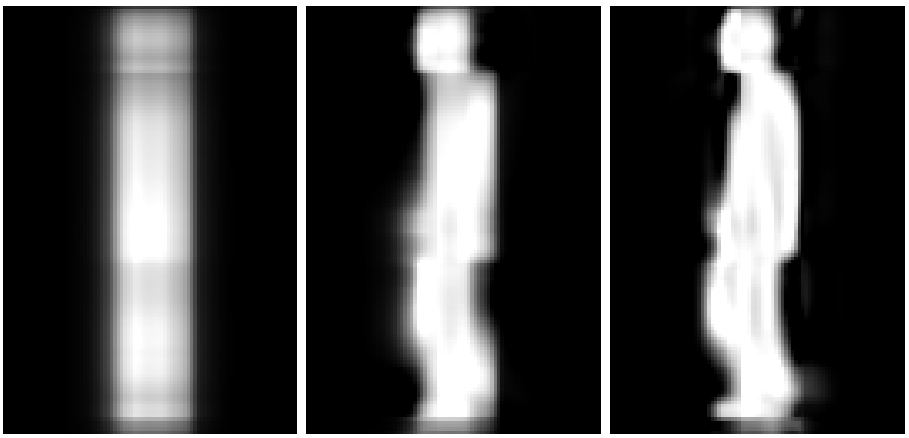
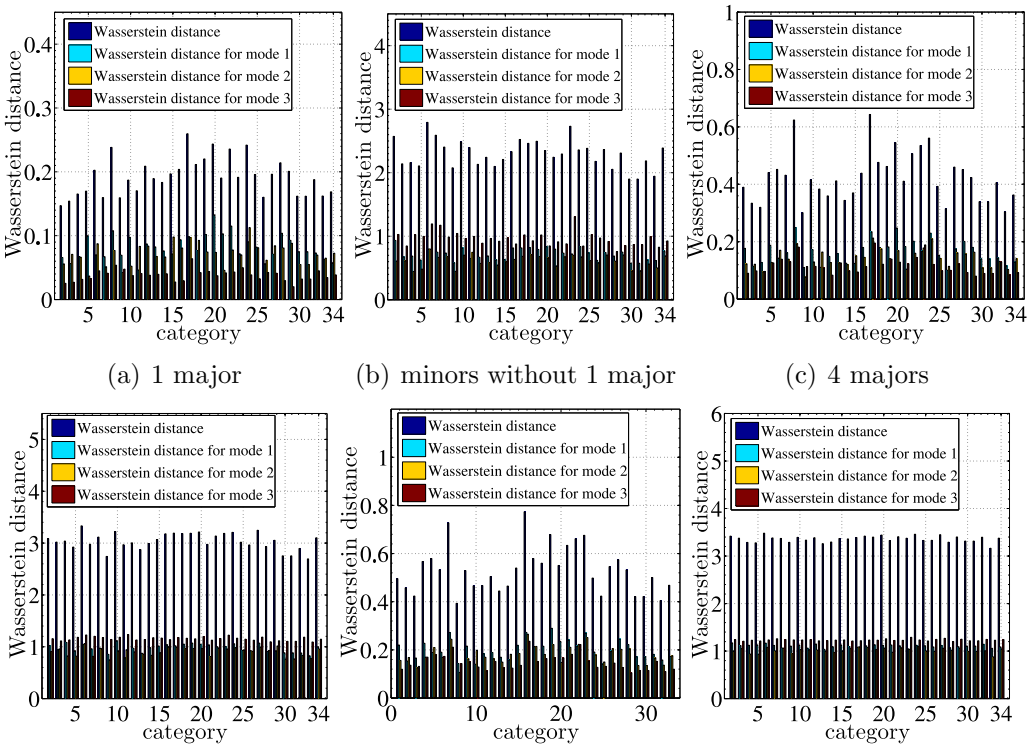


Figure 4.30: Wasserstein distances between substances of the first and kth categories. For the computation of the distances, we use 64, 57 and 20 eigenvectors of a category's tensor subspace for 1, 2 and 3 modes, respectively.



(a) (b) (c)

Figure 4.31: A first frame from the sequence reconstructed from principal components obtained by third-order principal component analysis. (a) reconstructed frame from one major principal component of each mode. (b) reconstructed frame from four major principal components of each mode. (c) reconstructed frame from ten major principal components of each mode.



(a) 1 major (b) minors without 1 major (c) 4 majors

(d) minors without 4 majors (e) 10 majors (f) minors without 10 majors

Figure 4.32: Wasserstein distances for sets of major and minor eigenvectors. (a), (c) and (e) show the Wasserstein distances between sets of major eigenvectors. (b), (d) and (f) show the Wasserstein distances between sets of minor eigenvectors.

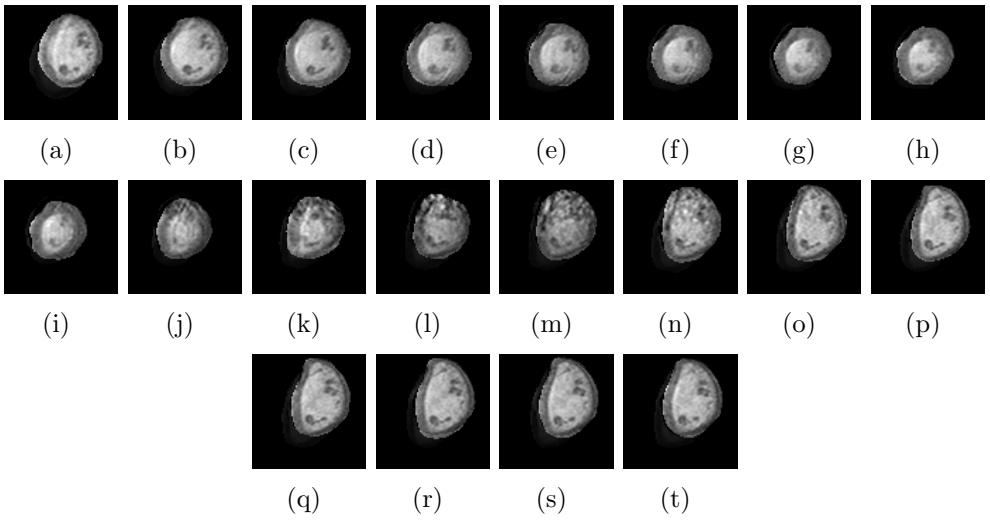


Figure 4.33: Sequence of beating heart. (a)-(t) show a sequence of a beating heart.

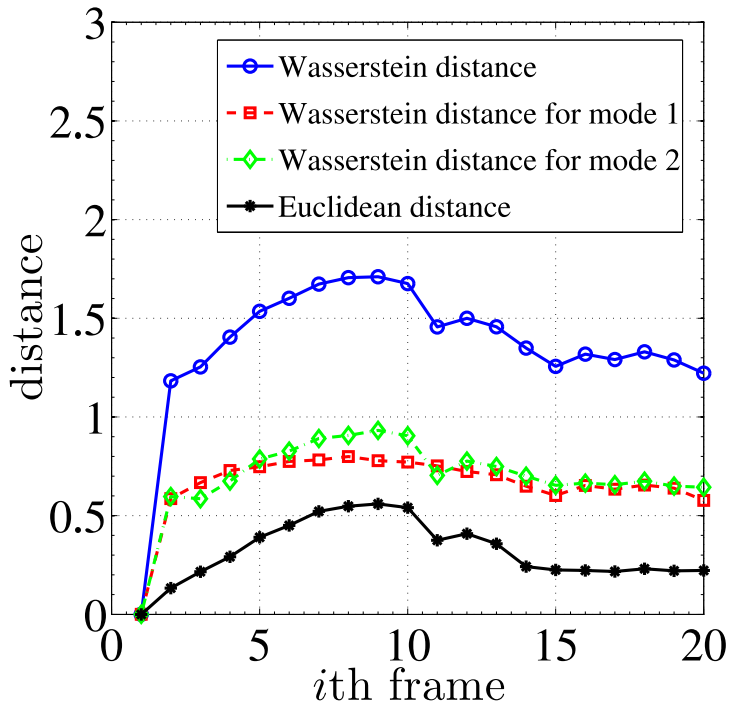


Figure 4.34: Wasserstein distances and Euclidean distance between first and ith frames for $i = 1, 2, \ldots, 20$. Plotted Euclidean distance is the relative distance for the L_2-norm of the first frame. T This relative distance is defined by $\|\boldsymbol{X}_1 - \boldsymbol{X}_i\|_{\mathrm{F}}/\|\boldsymbol{X}_1\|_{\mathrm{F}}$, where $\boldsymbol{X}_1$ and $\boldsymbol{X}_i$ are the first and ith frames, and $\|\cdot\|_{\mathrm{F}}$ is Frobenius norm.

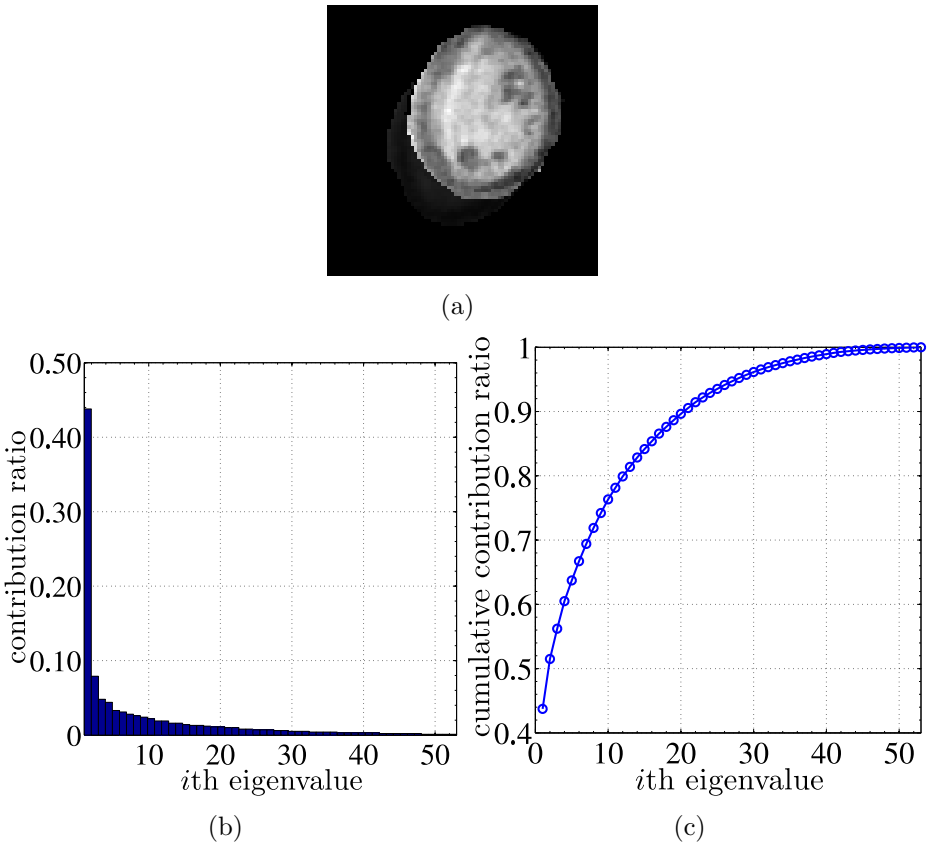


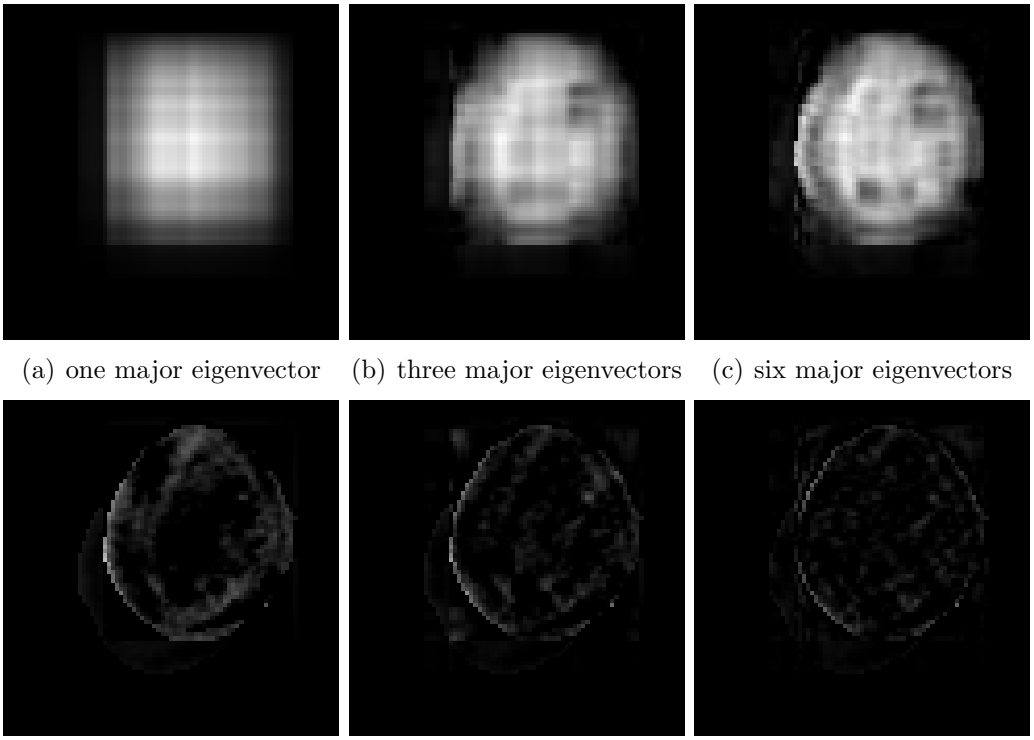
Figure 4.35: Example of an decomposition of an image. (a) first frame. (b) contribution ratio of eigenvalues. (c) cumulative contribution ratio of eigenvalues.

Third, we examine relation among eigenvectors of two images. We use successive two frames in a sequence shown in Fig. 4.37. By using singular value decomposition, we compute eigenvectors of mode-1 and -2 for the two frames. For these eigenvectors, we compute inner products among them. Figure 4.38 summarises the inner products. In Figs. 4.38 (a) and (b), from the first to third eigenvectors, the eigenvectors of the same order for two frames are almost coincident. Furthermore, we compute the difference between reconstructed images. Figure 4.39 shows the differences between the images reconstructed from selected eigenvectors. Compared to the Fig. 4.39, in Figs. 4.39, the differences between frames reconstructed from one, three and six major eigenvectors are not correct. On the other hand, the difference between frames reconstructed from minor eigenvectors are close to the difference between original successive frames. These results imply that analysis of minor components is important for analysis of time sequence data.

Forth, we compute Wasserstein distance between the first and ith frames by using major and minor eigenvectors. Figure 4.40 summarises the results of the computation of the Wasserstein distance. Figure 4.41 summarise the results of the computation of the Wasserstein distance for each mode. In Fig. 4.41, the difference between two frames are mainly depends on the second, third, forth, fifth and sixth eigenvectors. These eigenvectors represents change of successive frames. Using these eigenvectors, we can reconstruct the part around boundary of a slice image as shown in Fig 4.36. Furthermore, in Fig. 4.41, the Wasserstein distance for mode-2 is larger than the one for mode-1. This result imply that the change between the successive frames are represent by eigenvectors for mode-2.

Fifth, we compute the Wasserstein distance between subspaces of different categories. Using tensor principal component analysis for second-order tensors, we compute tensor subspaces of sequences of beating heart for different persons. Using these tensor subspaces, we compute the Wasserstein distances between subspaces of the first and kth categories. Figure 4.42 summarises these Wasserstein distances for all modes and each mode. Figures 4.42 (b) and (c) show that the differences among subspaces are expressed by principal components from the first to the sixth, since the differences of distances among subspaces are small if we use 47 minor eigenvectors. Figure 4.43 shows the Wasserstein distance for major and minor principal components. In many the distances for the categories in Figs. 4.42 (a) and Figs 4.43 (a)-(b), the distance for mode-2 is larger than the one for the other modes.

Sixth, we compute Wasserstein distance between subspaces of different categories by using the eigenvectors obtained from third-order principal component analysis. Figure 4.44 shows the distance between the first and kth subspaces. In Fig. 4.44, the distances for mode-2 mainly express the differ-



(a) one major eigenvector (b) three major eigenvectors (c) six major eigenvectors

(d) 30 minor eigenvectors (e) 28 minor eigenvectors (f) 25 minor eigenvectors

Figure 4.36: Reconstruction of the first frame. In (a), (b) and (c), the first frame is reconstructed by using one, three, six major eigenvectors for mode-1 and -2. In (d), (e) and (f), the first frame is reconstructed by using 52, 50 and 47 minor eigenvectors for mode-1 and -2.

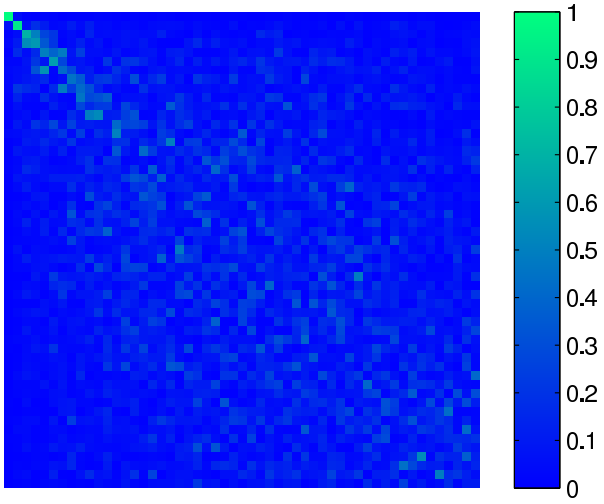
(a) (b) (c)

Figure 4.37: Successive two frames in a sequence. (a) Pre frame. (b) Post frame. (c) The difference of two frames. For visualisation, each pixel of the difference is displayed in its absolute value.

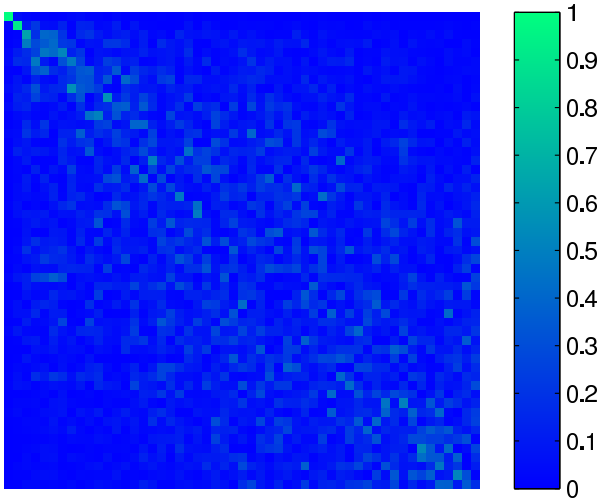
ence between subspaces.

Seventh, we reconstruct sequences of a beating heart by using only one, three and six principal components for mode-1, -2 and -3. Figure 4.45 shows the first frames of reconstructed sequences by using one, three and six principal components, respectively. Compared with the reconstruction from principal components obtained by second-order principal component analysis shown in Fig. 4.36, the slices of reconstructed sequences are blurred since the third-order principal component analysis seek bases for all frames in a sequence while singular value decomposition seek only bases for a frame. Figure 4.46 summarises the reconstructed sequences by using volume rendering of the sequences. In Fig. 4.46 (a), one major principal component express mean for each modes. In Fig. 4.46 (b), additional three major principal component express changes for direction of width, height and times. In Fig. 4.46 (c), additional 6 principal components express the changes along the directions of second-order derivatives.

Eighth, we compute the Wasserstein distances for major and minor principal components. Figure 4.47 summarises the the distances for only major principal components and for only minor principal components. In Fig. 4.47 (f), the variance of distances are smaller compared with the one in Figs. 4.47 (a), (c) and (d). These results imply that the difference among subspaces depend on the principal components from the first to tenth in each mode.

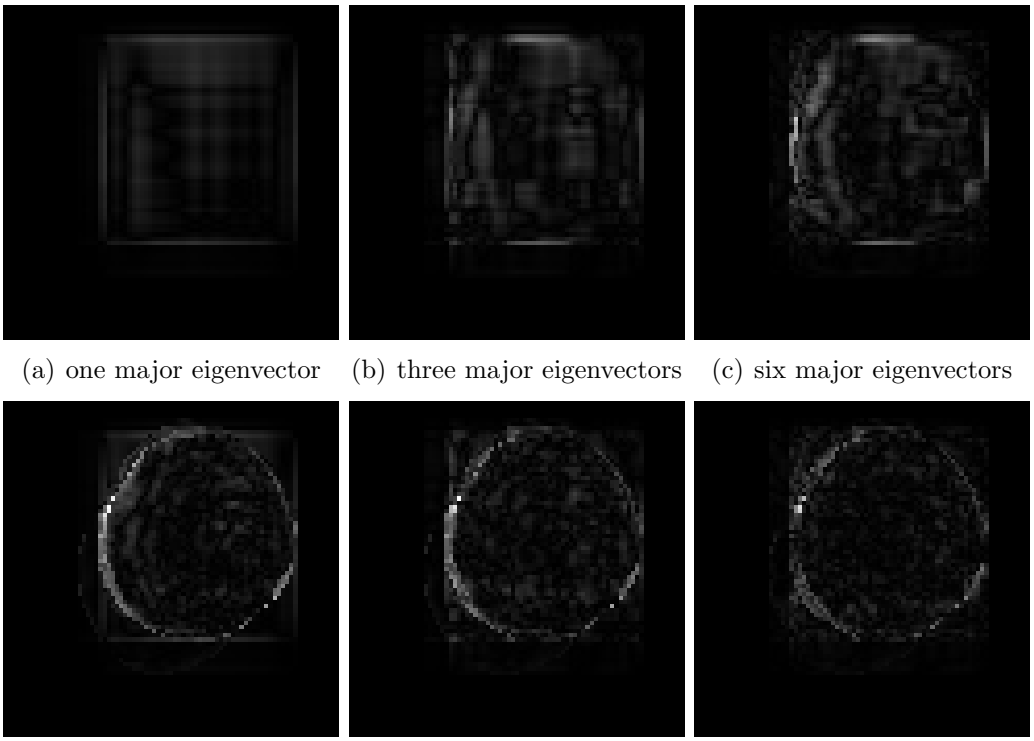


(a) mode 1



(b) mode 2

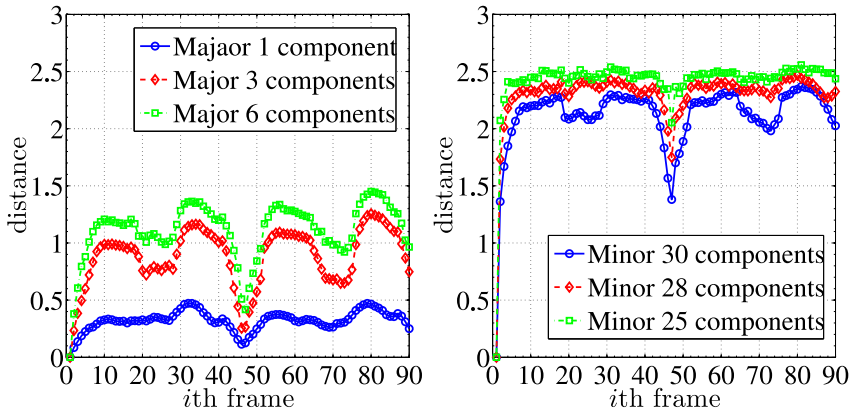
Figure 4.38: Absolute value of inner products for eigenvectors between 1st and 2nd frames. (a) and (b) show the inner products for eigenvectors of mode 1 and 2, respectively. In (a) and (b), from top to bottom, rows represent the eigenvectors of the second frame in descending order of eigenvalues. In (a) and (b), from left to right, columns represent the eigenvectors of the first frame in descending order of 53 eigenvalues.



(a) one major eigenvector (b) three major eigenvectors (c) six major eigenvectors

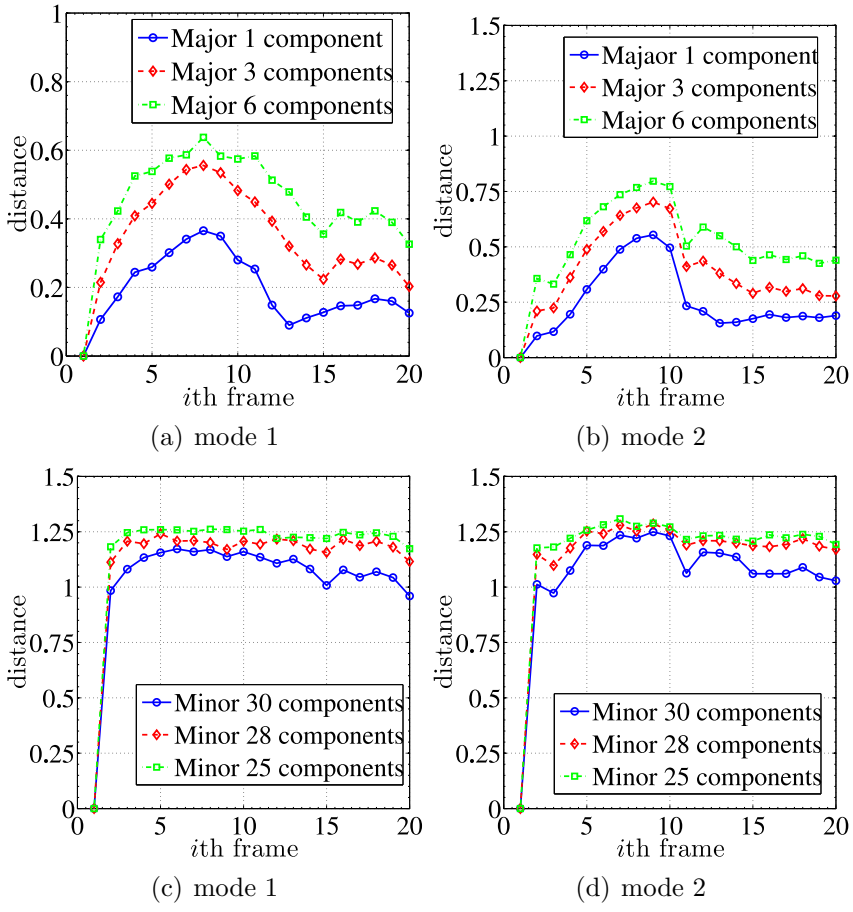
(d) 30 minor eigenvectors (e) 28 minor eigenvectors (f) 25 minor eigenvectors

Figure 4.39: Difference between reconstructed images.



(a) Major principal components (b) Minor principal components

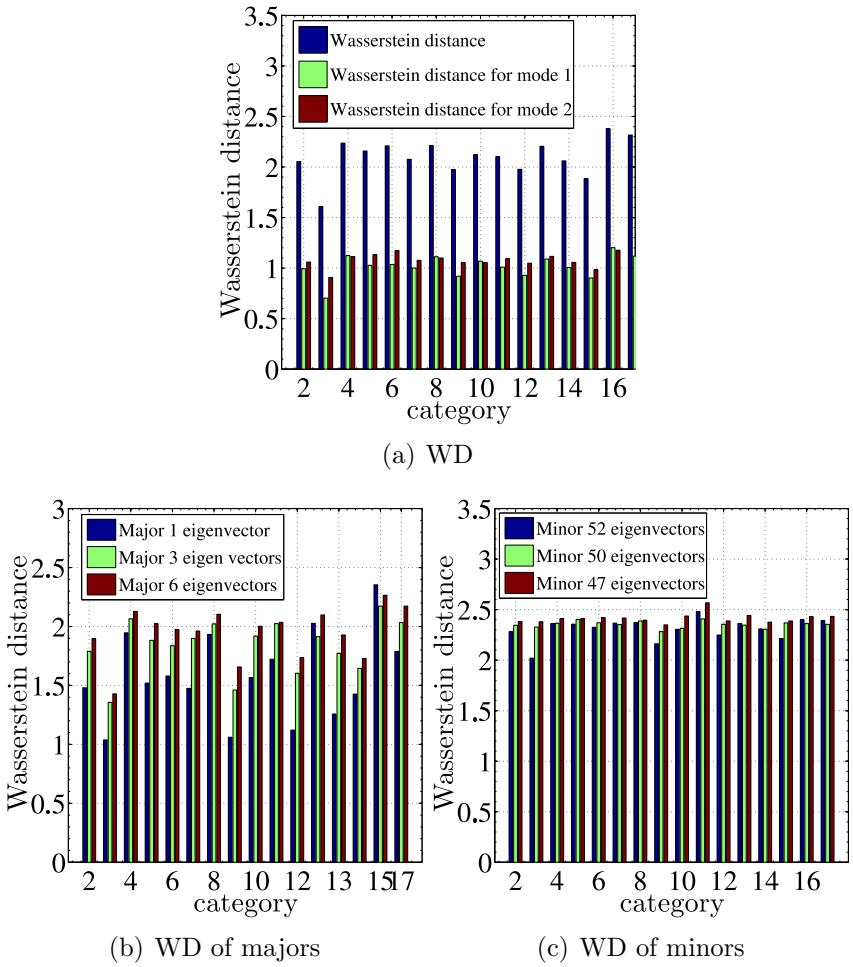
Figure 4.40: Wasserstein distances between first and ith frames for major and minor principal components.



(a) mode 1 (b) mode 2

(c) mode 1 (d) mode 2

Figure 4.41: Wasserstein distances for mode-1 and -2. Top and bottom rows summarise the Wasserstein distances computed by using major and minor principal components, respectively.

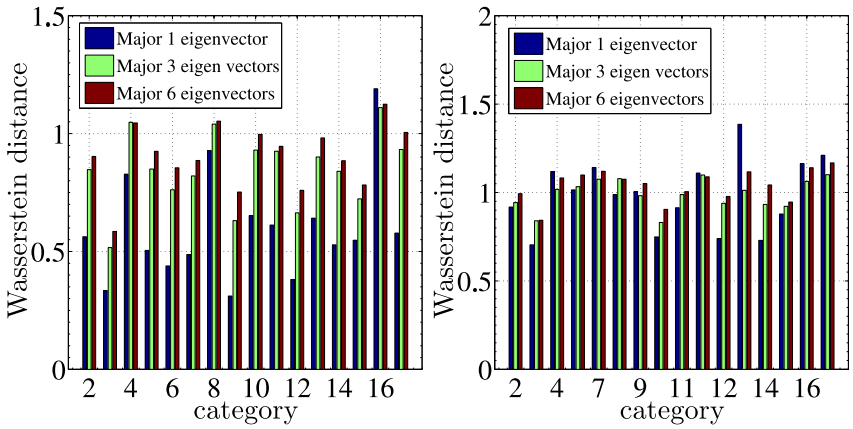


(a) WD

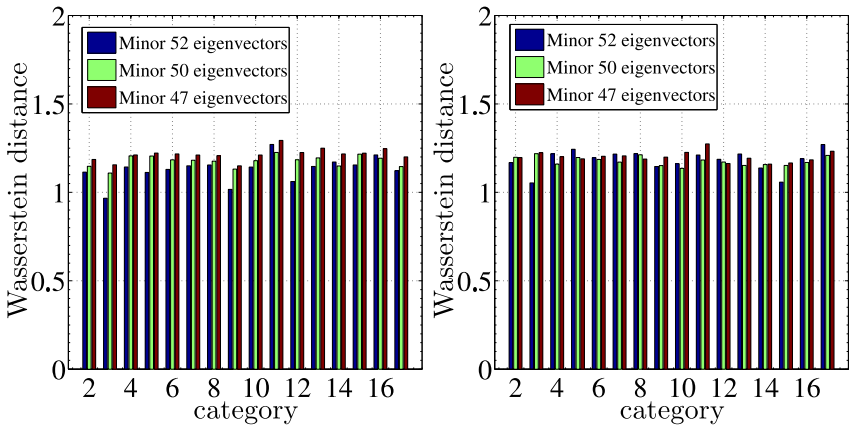
(b) WD of majors

(c) WD of minors

Figure 4.42: Wasserstein distances between first and ith frames for major and minor principal components.



(a) mode 1 (b) mode 2



(c) mode 1 (d) mode 2

Figure 4.43: Wasserstein distances for major and minor eigenvectors. (a), (c) and (e) show the Wasserstein distances between sets of major eigenvectors. (b), (d) and (f) show the the Wasserstein distances between sets of minor eigenvectors.

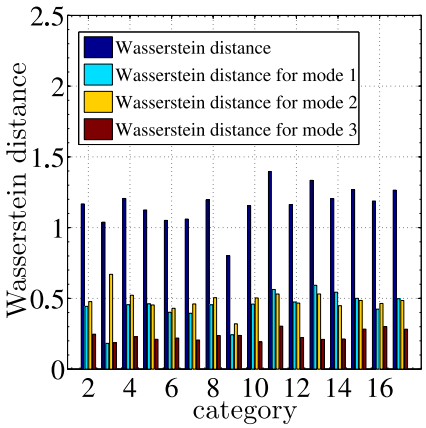
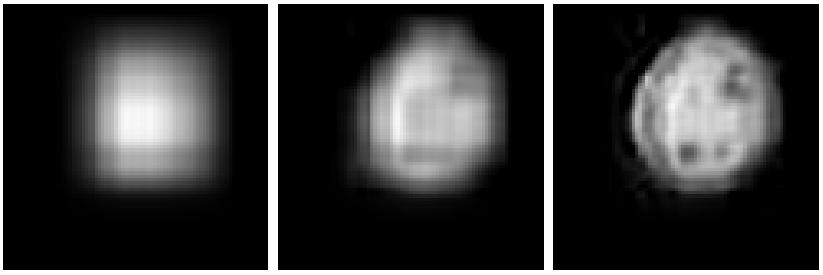
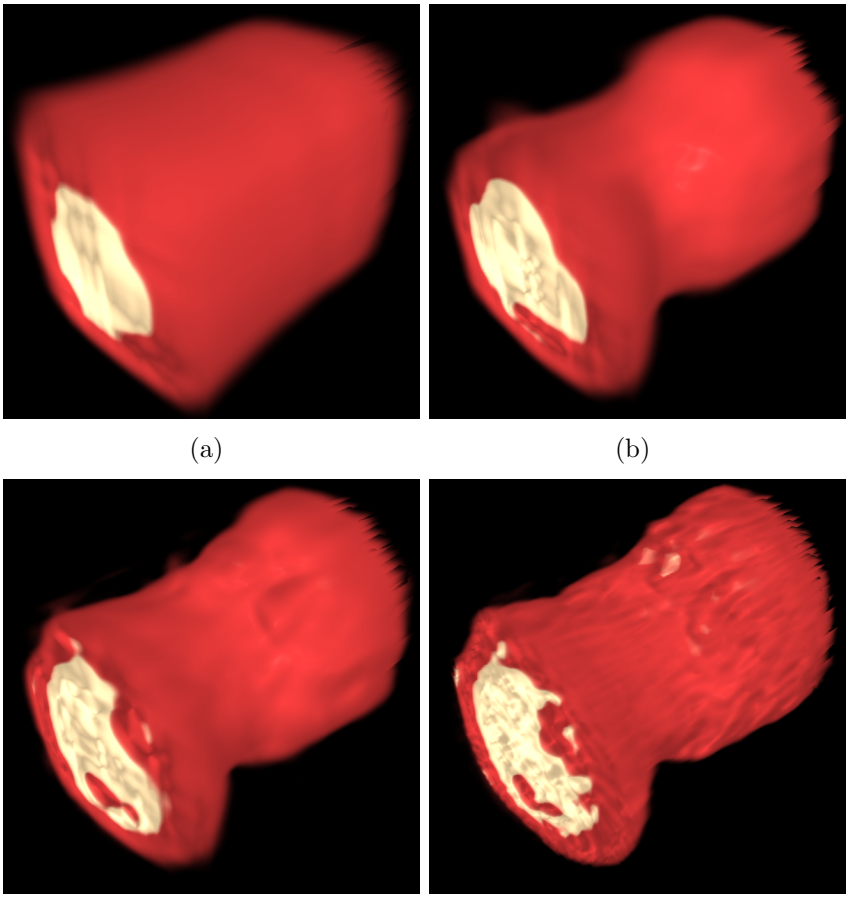


Figure 4.44: Wasserstein distances between first and ith frames computed by using 3rd-order-tensor representation.



(a) (b) (c)

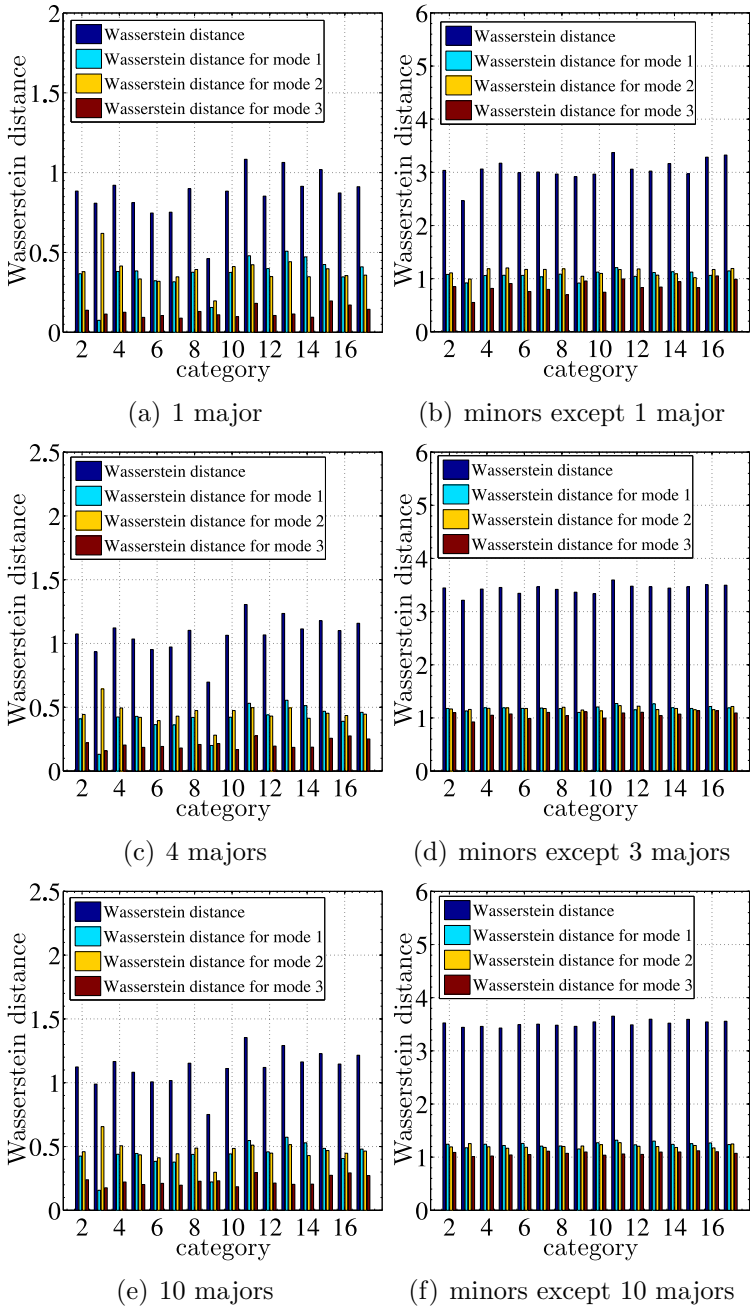
Figure 4.45: The first frames from the sequences reconstructed from principal component obtained by third-order principal component analysis. (a), (b) and (c) are the first frames of the sequence reconstructed by one, four and ten major principal components, respectively.



(a) (b)

(c) (d)

Figure 4.46: Volume rendering of reconstructed and original sequences. In (a)-(d), faces at left down part of images, on which white part exist, are the first frame of sequences. That is, these part in (a), (b), (c), and (d) coincident to the images shown in Figs. 4.36 (a), (b) and (c), and Fig. 4.35 (a), respectively. From the left down part to right up part, slices of a sequence are placed in times series. For these display, the voxel size for the direction of times series are multiplied by four since the data array is $81 \times 81 \times 20$ voxels. For these rendering, voxel size for time axis are multiplied by 4.



(a) 1 major (b) minors except 1 major

(c) 4 majors (d) minors except 3 majors

(e) 10 majors (f) minors except 10 majors

Figure 4.47: Wasserstein distances for mode-1 and -2. Top and bottom rows summarise the Wasserstein distances by using major and minor principal components, respectively.

Discussions

From the all numerical experiments, we have the following results of analysis.

- Wasserstein distance gives the similar distribution of distance to one of given by Euclidean distance but emphasis difference if we use all the principal component for the computation.
- Wasserstein distance measures the difference of subspaces in addition to difference of images.
- For the analysis of changes in an image sequence, the minor eigenvectors except 1 major eigenvector obtained by both the second-order and third-order principal component analysis express the difference between successive frames.
- Principal component eigenvectors from the second to sixth eigenvectors obtained by the second-order principal component analysis express the difference of subspaces of categories.
- Principal component eigenvectors from the second to tenth eigenvectors obtained by the third-order principal component analysis express the difference of subspaces of categories.

4.9 Summary

In this chapter, we first introduced tensor subspace method for Nth-order tensors. We then defined a tensor subspace of queries and mutual tensor subspace method for Nth-order tensors. In addition to the subspace methods, we define the Wasserstein distance between tensors. For defined subspace methods, we evaluated the accuracy of recognition by these two classification methods for gait data, volumetric data and spatio-temporal data. Furthermore, in experiments, we evaluated the effects of dimension reduction on tensor pattern recognition. Finally, we examined the geometry of multilinear subspace for pattern changes by using the Wasserstein distances.

For dimension reduction methods, this chapter adopted the equivalence between Nth-order tensor principal component analysis and N-dimensional singular value decomposition for $N = 3$, which introduced in Chapter 2. Furthermore, in accordance with section 3 of Chapter 2, we adopted the N-dimensional discrete cosine transform as an approximation of N-dimensional singular value decomposition.

In experiments, we first presented two validations for sequences of the two-dimensional images, and voxel images of human livers. Using the sequences

of binary images compressed by the iterative algorithm of the higher-order singular value decomposition and the three-dimensional discrete cosine transform, we computed the cumulative contribution ratio of the eigenvalues of a tensor subspace as the first validation.

As the second validation, using the sequence images and voxel images, we computed the accuracy of tensor pattern recognition for tensors compressed by the iterative algorithm and the three-dimensional discrete cosine transform for the sequences and volumetric data.

All the results in these two validations show the equivalent performance of the higher-order singular value decomposition and three-dimensional discrete cosine transform for third-order tensor pattern recognition. Furthermore, for the decomposition procedure, the results showed that tensor projection is independent of the order of selection of the modes in tensor projections. Moreover, for the sequences compressed the higher-order singular value decomposition, these results showed that the cumulative contribution ratio and recognition ratio are independent of the number of iterations in the decomposition procedure.

These numerical examples illustrated that the N-dimensional discrete cosine transform can be an acceptable approximation for N-dimensional singular value decomposition for 3 in tensor pattern recognition if we adopt the Euclidean distance as the metric of the pattern space. These examples also imply that the approximation of the higher-order singular value decomposition by the discrete cosine transform may be valid for tensors with order higher than the third order without an iterative computation method. These approximations are useful for the practical and fast computation of tensor recognition.

We, then, compared the recognition accuracy of tensor subspace method and mutual tensor subspace methods for cardiac MRI dataset. A sequence of volumetric data of human heart include three-dimensional geometrical change while heart beating. The results show that the mutual tensor subspace method achieve more robust than the tensor subspace method, since geometrical changes of beating heart represented by both category tensor subspace and query tensor subspace.

Finally, we experimentally examined the geometry of multilinear structure in tensor principal component analysis. Using the sequence of silhouette images, we extracted principal components and minor components. Visualised principal components show that major components represent outline shapes on images while minor component represent boundaries on images. By using the Wasserstein distance between tensors, we quantitatively showed the geometrical difference between sequential images. Using only selected major and minor eigenvectors of tensors, we explored which principal components

represent differences among sequential images. Furthermore, by applying the Wasserstein distances to compute the difference between tensor subspaces, we quantitatively showed the geometrical difference between two tensor subspaces that represent different two time sequences. The results clarified that temporal difference among images in time series are represented by minor components. These properties imply that we can quantitatively measure the difference among patterns by measuring changes in minor components of them. On the other hand, these results clarified that difference among tensor subspaces of categories are represented by major components. These results imply that the Wasserstein distance is applicable for practical computation of dissimilarity in the mutual tensor subspace method. Furthermore, for robust classification, we can explore the tensor subspace that mostly contribute to representation of difference among categories by using the Wasserstein distance.

Chapter 5

Feature Extraction

This Chapter is based on Publication of Internatinoal Confence "5. Discriminative Properties in Directional Distributions for Image Pattern Recognition".

5.1 Feature Extraction Methods

The gradient field of an image is a fundamental feature for image analysis. The gradient of each point is used as feature for boundary detection as a preprocessing for segmentation. Once segments in an image are robustly and accurately extracted the main process is stably achieved.

A well known gradient-based segment-detection operators are the Sobel, Laplacian and Canny operators [47, 67, 31]. These operations detect gradient of each point. The eigenvectors of the structure tensor [18] are used for robust computation of the directional gradient. For the robust computation, the average of the structure tensors in the neighbourhood of each point is adopted. The histogram of gradients evaluates the directional gradients of each point using the gradients in neighbourhoods of the point.

In this chapter, we clarify mathematical properties of histogram-based method for the gradient detection in image analysis. The histogram of oriented gradient (HOG) method [41] first construct the gradient histogram of each point as the feature of image pattern. They apply traditional discrimination techniques such as support vector machine for image pattern recognition. The method is a promising method for pedestrian detection [41, 182, 16]. Figure 5.1 shows this pipeline of the histogram of oriented gradient method. We clarify mathematical properties of the gradient histogram using the tensor-based feature expression and directional statistics.

The tensor is a mathematical tool for the expression of multidimensional

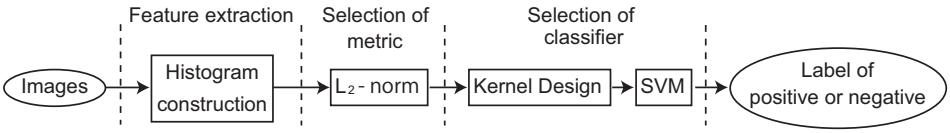


Figure 5.1: Pipeline of the HoG method. The pipeline consists of three steps: feature extraction, selection of metric and classifier. At the first step of feature extraction, the HoG method generate signature from distribution of gradients in local regions. At the second step of selection of metric, the HoG method adopts L_2-norm. At the third step of selection of classifier, the HoG method adopts kernel support vector machine (SVM). In this step, the HOG method needs appropriate kernel design for kernel SVM for a problem.

data. The gradient histogram is a mode three tensor for a two-dimensional image, since the histogram defines an additional dimension for data expression. Therefore, for the discrimination of images using gradient histogram, we are required to use the tensor version of the methods developed in pattern recognition.

Directional statistics [121] analyses the distribution of the directions of gradients of an image in a Euclidean space. Gradients are fundamental geometrical features for segment boundary extraction. Segment boundaries are employed for the detection of a particular object, such as a pedestrian or car, from a scene or a sequence of scenes. To measure differences and similarities among these histogram-based features, we are required to use a well-defined metric for the distributions. For this purpose, we redefined a histogram as a cyclic probability density function by using directional statistics [121].

Rubner *et al.* [141] derived the earth mover's distance (EMD) from a bipartite graph matching to compute the distance between histograms extracted from images. Rabin *et al.* [139] proposed the circular EMD since most popular local features are based on a one-dimensional circular histogram. The transportation problem of the EMD is coincident to the one of Wasserstein distance, which computes distance between probabilistic distributions [174].

By using the Wasserstein distance [174], we introduce a distance in directional statistics. Furthermore, we develop three methods to construct a histogram from a distribution of gradients. Combining these three construction methods and three aggregating methods for the local regions of an image, we define histogram extraction methods as feature extractions. directional-distribution-based features. Moreover, we explore the mathematical properties of the definition of the original HOG method. Finally, we evaluate the performances of the developed signatures and the original HoG

signature with the L_p-norm and the Wasserstein distance. Analysing the results of evaluations, we examine the mathematical properties required for accurate detection of objects in a scene/scenes. Table 5.1 summarises abbreviations of features and distances that we compared.

5.2 Related Works

Object recognition is the task of extracting and recognising a specific object/objects from an image/images or a video sequence/sequences. Feature-based [41, 182, 46, 110, 165] and appearance-based [129, 98] methods are the two main methodologies in object recognition. In feature-based methods, typical feature descriptors are object boundary contours [41, 182, 110, 165] and local colours [46, 45, 86] on the object. For recognition, features that describe segment boundaries are represented as elements in a pattern space. In ref. [110], a scale-invariant feature transform (SIFT) feature was proposed by Lowe. Extracted SIFT features are described as segment boundaries by histograms. Dalal and Triggs [41] proposed an HOG method for pedestrian detection. The HOG method detects objects with sharp boundaries and a uniform background, such as pedestrians and cars on pavements and streets, respectively, with high accuracy. Eleven years before the HoG method, Wakabayashi *et al.* [171] had been proposed a feature extraction method based on the histogram of gradients for character recognition. In this extraction method, contour detection and Gaussian filtering are used as preprocessing and postprocessing, respectively, for the construction of the histogram of gradients. Dollár *et al.* [46] combined the feature in the HOG method and multichannel features for the development of accurate pedestrian detection from multichannel images. Combinations of a multichannel feature and a bag-of-visual-words (BoW) feature are presented [165]. The recognition for these features is achieved by modern discrimination methods, such

Table 5.1: Glossary of abbreviations.

HoG	:	Histogram of gradients
SDD	:	Simple directional distribution
DD	:	Directional distribution
DDD	:	Dominant directional distribution
WD	:	Wasserstein distance
1WD	:	1-Wasserstein distance
B1WD	:	Binomial-distribution-based 1-Wasserstein distance
EMD	:	Earth Mover's distance

as the support vector machine (SVM) [163] and decision trees [46]. The appearance-based methods use the whole distribution of pixel values while many feature-based methods extract object boundary contours. Murase and Nayar [129] proposed a parametric eigenspace method for object recognition. By constructing an eigenspace of stored object images, their method achieved robust recognition against pose and illumination changes. However, their method focuses on only cropped object images. Lampert *et al.* [98] proposed an appearance-based model for efficient object detection that finds only candidate locations for an object.

For object recognition, the segment boundaries of an object are extracted from a scene. For the extraction of object boundary contours, the gradient field is a fundamental feature. As basic operations in image processing, several edge detectors, which are based on the gradient operation, have been proposed [47, 67, 31]. Several pipelines from segmentation to recognition through feature extraction have been proposed [41, 182, 46]. In the segmentation stage, gradient-based edge detection is an essential procedure. The formulation of the segmentation problem as an energy minimisation problem has been one of most commonly used techniques [48]. For the minimisation problem, Mumford and Shah proposed energy function adopting information of gradients of an image [128]. For the optimisation of the Mumford-Shah functional, El-Zehiry and Grady proposed combinatorial optimisation [48]. Dalal and Triggs explained that this combination of the SVM and learning data selects contour boundaries from the extracted feature for recognition [41]. Vondrick *et al.* [170] developed an algorithm to recover an image from an extracted HoG feature. They claimed that by visualising the HoG feature, we can gain a more intuitive understanding of the detection system. Their claim implies that the important mathematical properties of the HoG feature in object recognition are still unclear.

For the matching of histograms, variant of the EMD have been proposed. Pele and Werman introduced a variant of the EMD for the matching of SIFT features [134]. Pele and Werman [135] developed an algorithm for obtaining a robust family of EMDs with thresholded ground distances. For a large scale histograms, an efficient EMD was also developed in ref. [106] by simplifying the linear programming in EMD computation.

5.3 Mathematical Preliminaries

5.3.1 Function Space

We assume that an image $f(\boldsymbol{x})$ is an element of Sobolev space $W^{2,1} \cap W^{2,2}$. When $f \in W^{2,1}$ and $f \in W^{2,2}$, f in a finite space $\Omega \in \mathbb{R}^2$ satisfies the conditions

$$\|f\|_{W^{2,1}} = \int_\Omega (|f| + |\nabla f|)\, d\boldsymbol{x} < \infty, \tag{5.1}$$

$$\|f\|^2_{W^{2,2}} = \int_\Omega \left(|f|^2 + |\nabla f|^2\right) d\boldsymbol{x} < \infty. \tag{5.2}$$

For the gradients of images f and g, we have the following theorem.

Theorem 5.1 *For functions f and g, iff $\nabla f = \nabla g$, then $f = g +$* constant.

For the normalisation by the L_1- and L_2norms, we have the following theorems.

Theorem 5.2 *Assuming $f \in L_1(\Omega) \cap L_2(\Omega)$ for a finite closed set Ω, we have the relation*

$$\|f(\boldsymbol{x})\|_1 \leq \sqrt{\Omega}\|f(\boldsymbol{x})\|_2. \tag{5.3}$$

(Proof) For all $f \in L_p$ using the Cauchy-Schwartz inequality, we have

$$\|f\|_1 = \int_\Omega |f(\boldsymbol{x})| \cdot 1\, d\boldsymbol{x} \leq \left(\int_\Omega |f(\boldsymbol{x})|^2 d\boldsymbol{x}\right)^{1/2} \left(\int_\Omega 1^2 d\boldsymbol{x}\right)^{1/2} = \sqrt{\Omega}\|\boldsymbol{x}\|_2. \tag{5.4}$$

(Q.E.D.)

Theorem 5.3 *The mapping $\phi : \frac{f}{\|f\|_1} \mapsto \frac{f}{\|f\|_2}$ is a nonlinear mapping.*

(Proof) If we assume that the transform ϕ is linear, that is,

$$\frac{f}{\|f\|_1} = \phi(\frac{f}{\|f\|_2}) = \int_\Omega K(\boldsymbol{x}, \boldsymbol{y}) \frac{f}{\|f\|_2} d\boldsymbol{y}, \tag{5.5}$$

where K is independent of f, then the operator K satisfies the relation

$$K(\boldsymbol{x}, \boldsymbol{y}) = \frac{\|f\|_2}{\|f\|_1} \delta(\boldsymbol{x} - \boldsymbol{y}) \tag{5.6}$$

for all $f \in H$. Since $\alpha(f) = \frac{\|f\|_2}{\|f\|_1}$ is a function of f, K depends on f. This property of K contradicts the assumption on ϕ, implying that ϕ is a nonlinear transform. (Q.E.D.)

From Theorem 2 and 3, we can infer that separation ratio for L_2-normalised functions is higher than that for L_1-normalised functions in the discrimination by L_1- and L_2-norms.

Using a window function $W(\boldsymbol{y})$ and constants $\alpha, \beta \geq 0$, $\alpha \neq 0 \cup \beta \neq 0$, we define L_1- and L_2-norms for cropped region of an image as

$$\begin{aligned} \|f\|_{1,W} &= \int_{\mathbb{R}^2} \left\{ \int_{W(\boldsymbol{y})} \left(\alpha|f(\boldsymbol{x})| + \beta|\nabla f(\boldsymbol{x})|\right) d\boldsymbol{x} \right\} d\boldsymbol{y} \\ &= \int_{\mathbb{R}^2} W(\boldsymbol{x}-\boldsymbol{y}) \left(\alpha|f(\boldsymbol{x})| + \beta|\nabla f(\boldsymbol{x})|\right) d\boldsymbol{x}d\boldsymbol{y}, \end{aligned} \tag{5.7}$$

and

$$\begin{aligned} \|f\|_{2,W}^2 &= \int_{\mathbb{R}^2} \left\{ \int_{W(\boldsymbol{y})} \left(\alpha|f(\boldsymbol{x})|^2 + \beta|\nabla f(\boldsymbol{x})|^2\right) d\boldsymbol{x} \right\} d\boldsymbol{y} \\ &= \int_{\mathbb{R}^2} W(\boldsymbol{x}-\boldsymbol{y}) \left(\alpha|f(\boldsymbol{x})|^2 + \beta|\nabla f(\boldsymbol{x})|^2\right) d\boldsymbol{x}d\boldsymbol{y}, \end{aligned} \tag{5.8}$$

respectively.

For gradients of an image if we set $\alpha = 0, \beta = 1$, we have L_1- and L_2-norms as

$$\|f\|_{1,W} = \int_{\mathbb{R}^2} W(\boldsymbol{x}-\boldsymbol{y})|\nabla f| d\boldsymbol{x}\boldsymbol{y}, \tag{5.9}$$

$$\|f\|_{2,W}^2 = \int_{\mathbb{R}^2} W(\boldsymbol{x}-\boldsymbol{y})|\nabla f|^2 d\boldsymbol{x}\boldsymbol{y}, \tag{5.10}$$

respectively.

We consider that the distribution of directions is a probabilistic distribution of directions. As the special case of the Sobolev-Wasserstein distance, we introduce the distance between probabilistic distributions of directions. The p-Wasserstein distance [174] between a pair of probabilistic distributions $f(x)$ and $g(y)$ for $x \in X$ and $y \in Y$ is

$$W_p(f,g) = \min_c \left(\int_X \int_Y |f(x) - g(y)|^p c(x,y) dxdy \right)^{1/p}, \tag{5.11}$$

where $c(x,y)$ is a cost function.

Furthermore, for two images $f(\boldsymbol{x}), \boldsymbol{x} \in X$ and $g(\boldsymbol{y}), \boldsymbol{y} \in Y$, using a constant $p \in \{1,2\}$, we define Sobolev-Wasserstein distance

$$W_p(f,g) = \min \int_X \int_Y \left(\alpha|f(\boldsymbol{x}) - g(\boldsymbol{x})|^p + \beta|\nabla f - \nabla g|^p\right) c(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{x}d\boldsymbol{y}, \tag{5.12}$$

where $c(\boldsymbol{x},\boldsymbol{y})$ is a cost function.

5.3.2 Directional Gradient and Structure Tensor

The directional gradient of an image $f(\boldsymbol{x})$ for $\boldsymbol{x} = (x, y)^\top$ in the direction of $\boldsymbol{\omega} = (\cos\theta, \sin\theta)^\top$ is computed as $D(\theta) = \frac{\partial f}{\partial \boldsymbol{\omega}} = \boldsymbol{\omega}^\top \nabla f$. The directional gradient[1] $D(\theta)$ evaluates the steepness, smoothness and flatness of f along the direction of vector $\boldsymbol{\omega}$.

For gradient field, we have the following proposition.

Proposition 5.1 *For f, setting the directional tensor $\boldsymbol{S} = \nabla f \nabla f^\top$, ∇f and $|\nabla f|^2$ are the eigenfunction u_1 and eigenvalue λ_1 of $\boldsymbol{S}$, respectively.*

Furthermore, for a local region $\Psi(\boldsymbol{c})$ defined around a point $\boldsymbol{c} \in \mathbb{R}^2$, we define a structure tensor

$$\bar{\boldsymbol{S}} = \frac{1}{|\Psi(\boldsymbol{c})|} \int_{\Psi(\boldsymbol{c})} \nabla f \nabla f^\top d\boldsymbol{x}. \tag{5.13}$$

For a structure tensor, we have the following proposition.

Proposition 5.2 *For a local regions $\Psi(\boldsymbol{c})$, the eigenvectors $\{u_i\}_{i=1}^2$ and eigenvalues $\{\lambda\}_{i=1}^2$ of a structure tensor $\bar{S}$ represent the average direction and magnitude, respectively, of a local region $\Psi(\boldsymbol{c})$.*

Figure 5.2(d) illustrates the distribution of structure tensors in a local region of an image. From Proposition 2, we can use a pair (λ_1, u_1) of the first eigenvalue and the first eigenfunction of a structure tensor $\bar{S}$ as the directional distribution of a local region $\Psi(\boldsymbol{c})$.

Using the L_2-norm, we define direction of gradient

$$\boldsymbol{n}(\boldsymbol{x}) = \frac{\nabla f(\boldsymbol{x})}{\|\nabla f(\boldsymbol{x})\|_2} = (\cos\theta, \sin\theta)^\top \tag{5.14}$$

and magnitude of gradient

$$m(\boldsymbol{x}) = \|\nabla f(\boldsymbol{x})\|_2, \tag{5.15}$$

we set $\theta = \angle(\nabla f)$. Using pairs of the directions of the gradients and their magnitudes in the regions Ω, we have the gradient field of an image as

$$\Phi(f) = \{d(f(\boldsymbol{x})) \mid \boldsymbol{x} \in \Omega\}. \tag{5.16}$$

For convenience in two-dimensional applications, the pair of the directional angle $\theta(\boldsymbol{x})$ and $m(\boldsymbol{x})$ such that $h(\boldsymbol{x}) = \langle \theta, m \rangle$ is used. Figures 5.2(b), (c)

[1]The census transform $c(\boldsymbol{x}) = \int_{|\boldsymbol{\omega}|=1} u(\boldsymbol{\omega}^\top \nabla f)\, d\boldsymbol{\omega}$, where $u(s)$ is the unit step function such that $u(s) = 1$ for $s \geq 0$ and $u(s) = 0$ for $s < 0$, locally evaluates the total steepness, smoothness and flatness of each point [64].

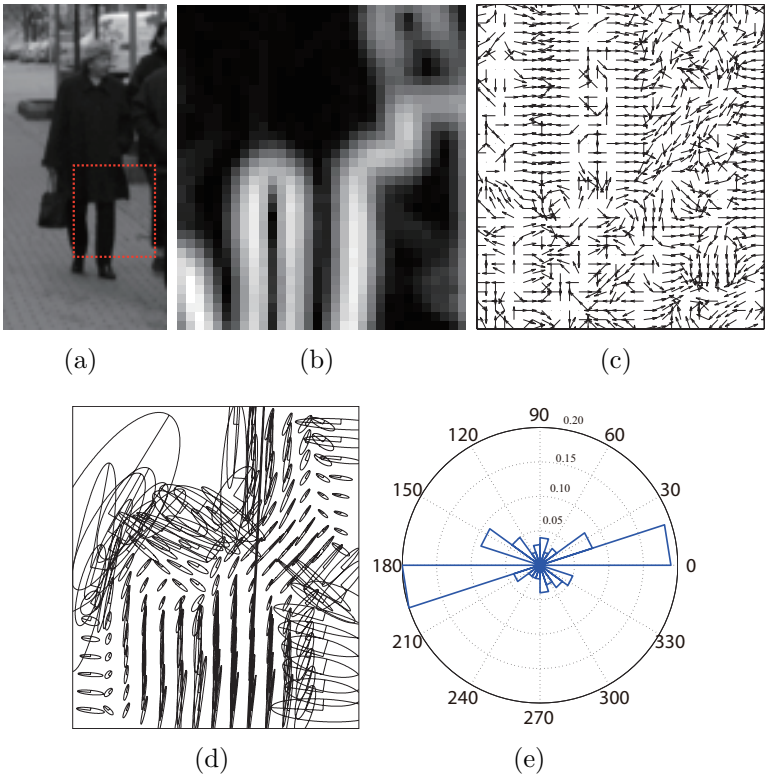


Figure 5.2: Example of directional statistics. (a) Grayscale image. (b) Magnitudes of gradients in a local region of the image. (c) Directions of gradients in a local region of the image. (d) Distribution of structure tensors in a local region of the image. (e) Circular histogram constructed with the gradient field.

and (e) show an example of distributions of magnitudes and directions of gradients, and its histogram for a local region of an image, respectively.

Using the directional gradients and direction $\boldsymbol{w} = (\cos\theta, \sin\theta)$, we can define the circular histogram

$$h(\theta) = \frac{\int_\Omega \boldsymbol{\omega}^\top \nabla f(\boldsymbol{x}) d\boldsymbol{x}}{\int_0^{2\pi} \int_\Omega \boldsymbol{\omega}^\top \nabla f(\boldsymbol{x}) d\boldsymbol{x} d\theta} \tag{5.17}$$

such that $h(\theta + 2\pi) = h(\theta)$ for an image $f(\boldsymbol{x})$. A histogram defined in eq (5.17) represents the gradient field.

For the histogram of directional gradients, we have the following theorem.

Theorem 5.4 *For the rotation operator R of rotation angle α, we define the rotated image $f(\boldsymbol{y})$, $\boldsymbol{y} = R\boldsymbol{x}$. For histograms $h(\theta)$ and $h'(\theta)$ of the original and rotated images, respectively, we have*

$$h'(\theta - \alpha) = h(\theta). \tag{5.18}$$

(Proof) We set the nabla operator $\nabla_{\boldsymbol{y}}$ and direction $\boldsymbol{w}_{\boldsymbol{y}}$ in the rotated coordinate system. For the directional gradients in the original and rotated coordinate systems, we have the relation

$$\boldsymbol{w}_{\boldsymbol{y}}^\top \nabla_y f(\boldsymbol{y}) = (R\boldsymbol{w})^\top \nabla_{\boldsymbol{y}} f(R\boldsymbol{x}) = \boldsymbol{w}^\top R^\top R \nabla f(\boldsymbol{x}) = \boldsymbol{w}^\top \nabla f(\boldsymbol{x}). \tag{5.19}$$

(Q.E.D.)

Using the pair (λ_1, u_1) of the first eigenvalue and first eigenvector of $\bar{S}$, the operation $\angle(\cdot)$ and the region Ω of an image, we construct the histogram

$$h_{\mathrm{D}}(\theta, \bar{s}(\boldsymbol{c})) = \int_{\boldsymbol{c}\in\Omega} \lambda_1 \delta(\theta - \angle(u_1)) d\boldsymbol{c}. \tag{5.20}$$

This histogram expresses the number of dominant directions for a direction θ in an image.

For practical computation of the histograms of directional gradients, for sampled image f_{ij} of discrete point $\boldsymbol{x}_{ij} = (i, j)^\top \in \mathbb{Z}^2$, we set region Ω around point $\boldsymbol{x}_{ij}$ and discrete angle $\theta_k = \frac{2\pi k}{n}, k = 0, 1, \ldots, n-1$ of directions $\boldsymbol{\omega}_k = (\cos\theta_k, \sin\theta_k)^\top$. For a sampled image $f_{ij} = f(\boldsymbol{x}_{ij})$ with discrete directions ω_k, we have a third-order tensor

$$\mathcal{H} = ((h_{ijk})), \quad h_{ijk} = \frac{\sum_\Omega \boldsymbol{\omega}_k \nabla f_{ij}}{\sum_{k=0}^{n-1} \sum_\Omega \boldsymbol{\omega}_k \nabla f_{ij}}, \tag{5.21}$$

which represents directional distribution on local region Ω around point $\boldsymbol{x}_{ij} = (i, j)^\top$. By introducing one parameter for a direction, we have a three-dimensional array from a two-dimensional array of a discrete image as a container of data.

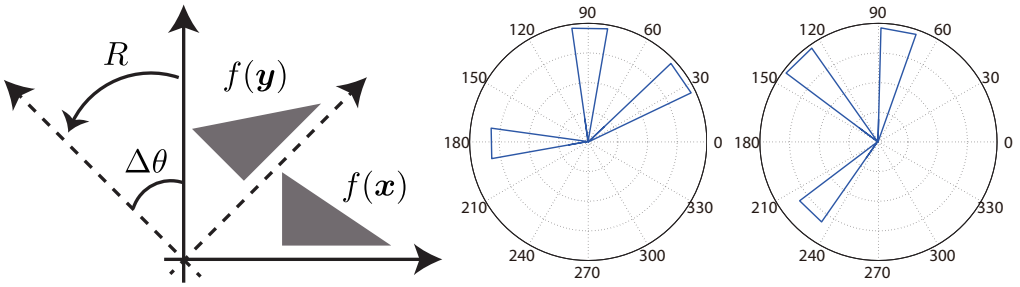


Figure 5.3: Rotation invariance in shape of circular histogram. (a) Original image $f(\boldsymbol{x})$ and rotated image $f(\boldsymbol{y})$. (b) Histogram $h(\theta)$ obtained from the original image. (c) Histogram $h'(\theta)$ obtained from the rotated image. The histogram in (c) is the histogram in (b) after rotation.

5.3.3 Aggregation Methods

To compute the directional distribution of a local region of an image, we define local regions using the same procedure as in ref. [41]. For a fixed point $\boldsymbol{c} \in \mathbb{R}^2$ on an image, positive constant $\alpha \in \mathbb{R}_+$ and positive integer $k \in \mathbb{Z}_+$, using a set of points $\boldsymbol{x} \in \mathbb{R}^2$ and the infinity norm $\|\cdot\|_\infty$, we define a local region and bounding box as

$$C(\boldsymbol{c}) = \{\boldsymbol{c} \,|\, \|\boldsymbol{x} - \boldsymbol{c}\|_\infty < \alpha\}, \quad B(\boldsymbol{c}) = \{\boldsymbol{x} \mid \|\boldsymbol{x} - \boldsymbol{c}\|_\infty < k\alpha\}, \tag{5.22}$$

which are called cells and blocks, respectively.

For the practical computation of cells and blocks, we divide a region of an image. For a constant α, we select a set of points $\{\boldsymbol{c}_{ij}\}_{i,j=1}^{M,N}$ with the conditions $C(\boldsymbol{c}_{ij}) \cap C(\boldsymbol{c}_{i'j'}) = \emptyset, (i,j) \neq (i',j')$ and $C(\boldsymbol{c}_{11}) \cup C(\boldsymbol{c}_{12}) \cup \ldots, C(\boldsymbol{c}_{MN}) = \Omega$. For $k = 2$, we select a set of blocks $B(\boldsymbol{c}'_l)$ with the condition $\boldsymbol{c}'_l \in \{\boldsymbol{\mu} | \boldsymbol{\mu} = (\boldsymbol{c}_{ij} + \boldsymbol{c}_{ij+1} + \boldsymbol{c}_{i+1j} + \boldsymbol{c}_{i+1j+1})/4\}$, $l = 1, 2, \ldots, (M-1)(N-1)$ so that each block consists of four cells.

Figures 5.4(a), (b) and (c) show the cases of no division, dividing by cells and dividing by blocks. By dividing an image and aggregating the divided regions, we obtain sets of cells and blocks, respectively. For these three dividing methods, we have three comparison methods as shown in Figs. 5.4(d), (e) and (f). The HoG method divides an image into cells, aggregates the cells as blocks and represents an image as a vector by connecting histograms of cells [2]. For vectors, cells, blocks and histograms, we have to select appropriate metrics.

[2]The scale-invariant feature transform divides an image into cells and represents an image as a vector connecting histograms of cells [110].

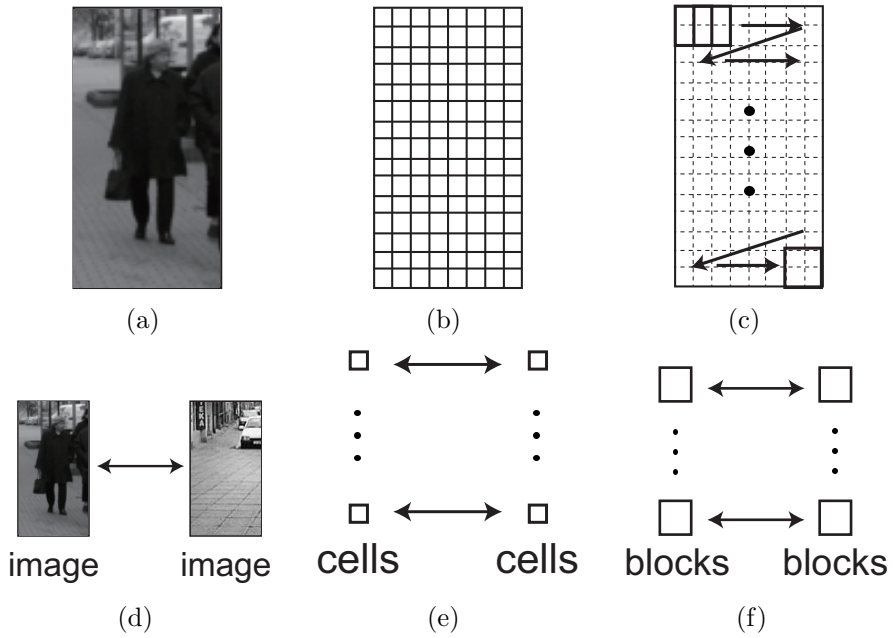


Figure 5.4: Methods of aggregating local regions and of measuring difference between aggregated local regions. (a) Whole region of an image with no division. (b) Cells dividing an image. (c) Cells aggregated into blocks. As shown in (d)-(f), we have three methods to discriminate the difference between two images. (d) Difference between images. (e) Sum of differences between cells. (f) Sum of differences between blocks.

5.3.4 Distribution of Directional Gradient

To compute the histograms from the distribution of directional gradients, we define local and global directional distributions for an image. Using only directions of gradients in local regions, we firstly define two histograms in local regions.

Definition 5.1 *For local regions $C(\boldsymbol{c})$ and $B(\boldsymbol{c})$, using the operation $\angle(\cdot)$, we define the histogram of simple directional distributions*

$$H^{\mathrm{C}}(\theta, \boldsymbol{c}) = \frac{\int_{C(\boldsymbol{c})} \delta(\theta - \angle(\nabla f)) d\boldsymbol{x}}{\int_0^{2\pi} \int_{C(\boldsymbol{c})} \delta(\theta - \angle(\nabla f)) d\boldsymbol{x}\, d\theta}, \tag{5.23}$$

$$H^{\mathrm{B}}(\theta, \boldsymbol{c}) = \frac{\int_{B(\boldsymbol{c})} \delta(\theta - \angle(\nabla f)) d\boldsymbol{x}}{\int_0^{2\pi} \int_{B(\boldsymbol{c})} \delta(\theta - \angle(\nabla f)) d\boldsymbol{x}\, d\theta}, \tag{5.24}$$

where δ is the Dirac delta function, respectively, as probabilistic distributions.

These histograms are the directional statistics [121] for a gradient field.

Furthermore, using the pairs of directions and magnitudes of gradients in local regions, we define two histograms.

Definition 5.2 *For local regions $C(\boldsymbol{c})$ and $B(\boldsymbol{c})$, using the operation $\angle(\cdot)$ and the magnitude $m(\boldsymbol{x})$ of a gradient at a point $\boldsymbol{x}$, we define the histograms of directional distributions*

$$H_{\mathrm{w}}^{\mathrm{C}}(\theta, \boldsymbol{c}) = \frac{\int_{C(\boldsymbol{c})} m(\boldsymbol{x}) \delta(\theta - \angle(\nabla f)) d\boldsymbol{x}}{\int_0^{2\pi} \int_{C(\boldsymbol{c})} m(\boldsymbol{x}) \delta(\theta - \angle(\nabla f)) d\boldsymbol{x}\, d\theta}, \tag{5.25}$$

$$H_{\mathrm{w}}^{\mathrm{B}}(\theta, \boldsymbol{c}) = \frac{\int_{B(\boldsymbol{c})} m(\boldsymbol{x}) \delta(\theta - \angle(\nabla f)) d\boldsymbol{x}}{\int_0^{2\pi} \int_{B(\boldsymbol{c})} m(\boldsymbol{x}) \delta(\theta - \angle(\nabla f)) d\boldsymbol{x}\, d\theta}, \tag{5.26}$$

as probabilistic distributions.

Here, the histogram $H_{\mathrm{w}}^{\mathrm{C}}(\theta, \boldsymbol{c})$ is the HoG descriptor (HoG glyph). This histogram represents the distribution of the magnitude of the gradients for each direction. These two local histograms represent the directional distributions of each local region of an image.

Moreover, we define histogram of global directional distributions.

Definition 5.3 *For an image, we define a normalised histogram of gradients over the image region $\Omega \in \mathbb{R}^2$ as*

$$H_{\mathrm{w}}^{\mathrm{G}}(\theta) = \frac{\int_{\boldsymbol{x} \in \Omega} m(\boldsymbol{x}) \delta\left(\theta - \angle(\nabla f)\right) d\boldsymbol{x}}{\int_0^{2\pi} \int_{\boldsymbol{x} \in \Omega} m(\boldsymbol{x}) \delta\left(\theta - \angle(\nabla f)\right) d\boldsymbol{x}\, d\theta}. \tag{5.27}$$

This histogram represents the distribution of the magnitude of the gradients for each direction.

5.3.5 Distribution of Dominant Directional Gradient

For the computation of the dominant directional distribution, we define the maximum histograms in a cell and block as

$$M^{\mathrm{C}}(\boldsymbol{c}) = \max_{\theta} H^{\mathrm{C}}_{\mathrm{w}}(\theta, \boldsymbol{c}), \quad M^{\mathrm{B}}(\boldsymbol{c}) = \max_{\theta} H^{\mathrm{B}}_{\mathrm{w}}(\theta, \boldsymbol{c}), \tag{5.28}$$

respectively. Using the maximum histograms in a cell and block, for a constant $\lambda > 0.5$, we define histograms of the dominant directions for a cell and block as

$$h^{\mathrm{C}}_{\mathrm{D}}(\theta, \boldsymbol{c}) = \begin{cases} H^{\mathrm{C}}_{\mathrm{w}}(\theta, \boldsymbol{c}), & \text{if } \ H^{\mathrm{C}}_{\mathrm{w}}(\theta, \boldsymbol{c}) \geq \lambda M^{\mathrm{C}}(\boldsymbol{c}) \\ 0, & \text{otherwise,} \end{cases} \tag{5.29}$$

$$h^{\mathrm{B}}_{\mathrm{D}}(\theta, \boldsymbol{c}) = \begin{cases} H^{\mathrm{B}}_{\mathrm{w}}(\theta, \boldsymbol{c}), & \text{if } \ H^{\mathrm{B}}_{\mathrm{w}}(\theta, \boldsymbol{c}) \geq \lambda M^{\mathrm{B}}(\boldsymbol{c}) \\ 0, & \text{otherwise,} \end{cases} \tag{5.30}$$

respectively. By assuming the directions in a local region change their direction and magnitude smoothly, the cut-off with maximum histograms finds dominant directions.

For a local regions, we define histograms of dominant directional distributions.

Definition 5.4 *For local regions $C(\boldsymbol{c})$ and $B(\boldsymbol{c})$ of an image, we have local histograms of the dominant directional distribution of*

$$H^{\mathrm{C}}_{\mathrm{D}}(\theta, \boldsymbol{c}) = \frac{h^{\mathrm{C}}_{\mathrm{D}}(\theta, \boldsymbol{c})}{\int_0^{2\pi} h^{\mathrm{C}}_{\mathrm{D}}(\theta, \boldsymbol{c}) d\theta}, \tag{5.31}$$

$$H^{\mathrm{B}}_{\mathrm{D}}(\theta, \boldsymbol{c}) = \frac{h^{\mathrm{B}}_{\mathrm{D}}(\theta, \boldsymbol{c})}{\int_0^{2\pi} h^{\mathrm{B}}_{\mathrm{D}}(\theta, \boldsymbol{c}) d\theta}, \tag{5.32}$$

respectively.

Furthermore, for a global region, we define histogram of dominant directional distribution.

Definition 5.5 *For the region Ω of an image, we have the global dominant directional distribution*

$$H^{\mathrm{G}}_{\mathrm{D}}(\theta) = \frac{\int_{\boldsymbol{c} \in \Omega} \int_{\boldsymbol{x} \in B(\boldsymbol{c})} h^{\mathrm{C}}_{\mathrm{D}}(\theta, \boldsymbol{x}) d\boldsymbol{x} d\boldsymbol{c}}{\int_0^{2\pi} \int_{\boldsymbol{c} \in \Omega} \int_{\boldsymbol{x} \in B(\boldsymbol{c})} h^{\mathrm{C}}_{\mathrm{D}}(\theta, \boldsymbol{x}) d\boldsymbol{x} d\boldsymbol{c} d\theta}. \tag{5.33}$$

5.3.6 Gradient-Based Discrimination

For a reference image f and a template image g, using a sliding window W, the gradient-based feature $F(f)$ of an image f and the metric D for features, gradient-based discrimination is achieved by minimising

$$J(f) = D(F(g_W(\boldsymbol{x})), F(f(\boldsymbol{x}))), \tag{5.34}$$

where

$$g_W(\boldsymbol{x}) = \chi_W(\boldsymbol{x})g(\boldsymbol{x}),\ \chi_W(\boldsymbol{x}) = \begin{cases} 1, & \text{if } \boldsymbol{x} \in W, \\ 0, & \text{otherwise,} \end{cases} \tag{5.35}$$

is a moving window on the image g. To establish high-accuracy discriminating, we need to construct a discriminative gradient-based feature $F(f)$ and select an appropriate metric for this gradient-based feature.

5.4 Feature Extraction and Discrimination

Table 5.2 summarises all the histograms defined in section 3 and the histogram used in the HoG feature. Using these histograms, we define feature extraction methods that are based on histograms of local and global directional distributions. Furthermore, we introduce methods to discriminate the difference between these features. Table 5.3 summarises all the pairs of features and discrimination criteria. Figure 5.5 summarises the feature extraction methods.

5.4.1 Local Directional Distribution Methods

Using the histograms of cells and blocks, we define three local features based on local directional distributions. For the histogram $H(\theta, \boldsymbol{c}) \in \{H^{\mathrm{C}}(\theta, \boldsymbol{c}), H^{\mathrm{B}}(\theta, \boldsymbol{c})\}$,

Table 5.2: Summary of histograms defined in section 3 and 4.3.

Histogram	Explanation	Definition
$H^{\mathrm{C}}(\theta, \boldsymbol{c})$	Local directional distribution in a cell	eq. (5.23)
$H^{\mathrm{B}}(\theta, \boldsymbol{c})$	Local directional distribution in a block	eq. (5.24)
$H^{\mathrm{C}}_{\mathrm{w}}(\theta, \boldsymbol{c})$	Local directional distribution with magnitudes in a cell	eq. (5.25)
$H^{\mathrm{B}}_{\mathrm{w}}(\theta, \boldsymbol{c})$	Local directional distribution with magnitudes in a block	eq. (5.26)
$H^{\mathrm{C}}_{\mathrm{D}}(\theta, \boldsymbol{c})$	Local distribution of dominant directions in a cell	eq. (5.31)
$H^{\mathrm{B}}_{\mathrm{D}}(\theta)$	Local distribution of dominant directions in a block	eq. (5.32)
$H^{\mathrm{G}}_{\mathrm{w}}(\theta, \boldsymbol{c})$	Global directional distribution in an image	eq. (5.27)
$H^{\mathrm{G}}_{\mathrm{D}}(\theta, \boldsymbol{c})$	Global distribution of dominant directions in an image	eq. (5.33)
$H_{\mathrm{H}}(\theta, \boldsymbol{c})$	Histogram that is normalised by each block in the HoG method	eq. (5.49)

Table 5.3: Summary of pairs of features and discrimination criteria. For the simple directional distribution (SDD), directional distribution (DD), dominant directional distribution (DDD) and histogram of oriented gradients (HoG), this table shows whether each discrimination method is available or not by ◯ and ×, respectively. Features are divided with respect to whether they are based on a local histogram or a global histogram.

region			Local							Global	
feature			SDD		DD		DDD		HoG	DD	DDD
histogram			H^{C}	H^{B}	$H_{\mathrm{w}}^{\mathrm{C}}$	$H_{\mathrm{w}}^{\mathrm{B}}$	$H_{\mathrm{D}}^{\mathrm{C}}$	$H_{\mathrm{D}}^{\mathrm{B}}$	H_{H}	$H_{\mathrm{w}}^{\mathrm{G}}$	$H_{\mathrm{D}}^{\mathrm{G}}$
discrimination	C method:	D_{Cell}	◯	×	◯	×	◯	×	×	×	×
	B method:	D_{Block}	×	◯	×	◯	×	◯	◯	×	×
	BWC method:	D_{BWC}	◯	×	◯	×	◯	×	×	×	×
	G method:	D_{G}	×	×	×	×	×	×	×	◯	◯
	L_p-norm:	D_{L_p}	◯	◯	◯	◯	◯	◯	◯	◯	◯

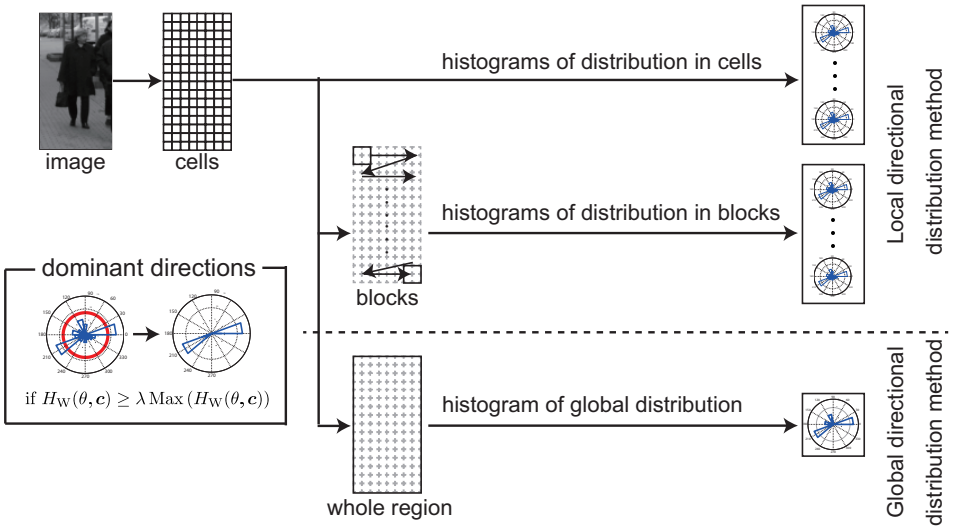


Figure 5.5: Flow of feature extraction methods. There are three flows. The top row shows feature extraction from the entire region of an image. The middle and bottom rows show feature extraction from cells and blocks, respectively. In the middle and bottom rows, a set of histograms is extracted. For these three extraction methods, we can adopt three different types of histograms. In other words, we can use the simple directional distribution, the directional distribution and the dominant directional distribution. The small box on the left summarises the extraction of the dominant directional distribution in each cell.

we define the simple directional distribution (SDD) feature as

$$F_{\mathrm{SDD}}(f) = \{P_s(\boldsymbol{c}) \mid \boldsymbol{c} \in \Omega\}, \quad P_s(\boldsymbol{c}) = \{H(\theta, \boldsymbol{c}) \mid 0 \leq \theta < 2\pi\}. \tag{5.36}$$

Furthermore, for the directional distribution $H_{\mathrm{w}}(\theta, \boldsymbol{c}) \in \{H_{\mathrm{w}}^{\mathrm{C}}(\theta, \boldsymbol{c}), H_{\mathrm{w}}^{\mathrm{B}}(\theta, \boldsymbol{c})\}$, we define the directional distribution (DD) feature as

$$F_{\mathrm{DD}}(f) = \{P_{\mathrm{w}}(\boldsymbol{c}) \mid \boldsymbol{c} \in \Omega\}, \quad P_{\mathrm{w}}(\boldsymbol{c}) = \{H_{\mathrm{w}}(\theta, \boldsymbol{c}) \mid 0 \leq \theta < 2\pi\}. \tag{5.37}$$

For the dominant directional distribution $H_{\mathrm{D}}(\theta, \boldsymbol{c}) \in \{H_{\mathrm{D}}^{\mathrm{C}}(\theta, \boldsymbol{c}), H_{\mathrm{D}}^{\mathrm{B}}(\theta, \boldsymbol{c})\}$, we define the dominant direction distribution (DDD) feature as

$$F_{\mathrm{DDD}}(f) = \{P_{\mathrm{D}}(\boldsymbol{c}) \mid \boldsymbol{c} \in \Omega\}, \quad P_{\mathrm{D}}(\boldsymbol{c}) = \{H_{\mathrm{D}}(\theta, \boldsymbol{c}) \mid 0 \leq \theta < 2\pi\}. \tag{5.38}$$

For these three extracted local features, we construct four methods to discriminate differences between images. Firstly, for the SDD, DD and DDD features extracted from cells, we define two discrimination methods. We set $F(f) = \{P(\boldsymbol{c}) | \boldsymbol{c} \in \Omega\}$ and $F(g) = \{Q(\boldsymbol{c}) | \boldsymbol{c} \in \Omega\}$ as the extracted features for images f and g, respectively. We define the cell-based discrimination method (**C method**), which computes the difference between image patterns using

$$D_{\mathrm{Cell}} = \int_{\boldsymbol{c} \in \Omega} \mathrm{D_W}\left(P(\boldsymbol{c}), Q(\boldsymbol{c})\right) d\boldsymbol{c}, \tag{5.39}$$

where $\mathrm{D_W}(\cdot, \cdot)$ is the p-Wasserstein distance or binomial p-Wasserstein distance. This discrimination method sums the Wasserstein distance between each corresponding pair of cells in two images.

Second, we define the blockwise-cell-based discrimination method (**BWC method**), which discriminates a pairs of image patterns using

$$D_{\mathrm{BWC}} = \int_{\boldsymbol{c} \in \Omega} \int_{\boldsymbol{x} \in B(\boldsymbol{c})} \mathrm{D_W}\left(P(\boldsymbol{x}), Q(\boldsymbol{x})\right) d\boldsymbol{x} d\boldsymbol{c}. \tag{5.40}$$

This discrimination method sums the Wasserstein distance between cells in each corresponding pair of blocks in two images. In eqs. (5.39) and (5.40), both $P(\boldsymbol{c})$ and $Q(\boldsymbol{c})$ are given by the histograms of cells $H^{\mathrm{C}}, H_{\mathrm{w}}^{\mathrm{C}}$ or $H_{\mathrm{D}}^{\mathrm{C}}$.

Third, for the SDD, DD and DDD features extracted from blocks, we define the block-based discrimination method (**B method**), which involves evaluating the criterion

$$D_{\mathrm{Block}} = \int_{\boldsymbol{c} \in \Omega} \mathrm{D_W}(P(\boldsymbol{c}), Q(\boldsymbol{c})) d\boldsymbol{c}, \tag{5.41}$$

where both $P(\boldsymbol{c})$ and $Q(\boldsymbol{c})$ are given by histograms of blocks $H^{\mathrm{B}}, H_{\mathrm{w}}^{\mathrm{B}}$ or $H_{\mathrm{D}}^{\mathrm{B}}$. This discrimination method sums the Wasserstein distance of each corresponding pair of blocks in two images.

Fourth, for the SDD, DD and DDD features extracted from cells and blocks, we define the L_p-norm. To define the L_p-norm for cells and blockwise cells, we set $P(\boldsymbol{c}) = \{H_P(\theta, \boldsymbol{c}) \mid 0 \leq \theta < 2\pi\}$ and $Q(\boldsymbol{c}) = \{H_Q(\theta, \boldsymbol{c}) \mid 0 \leq \theta < 2\pi\}$, which are given by the histograms H^{C}, $H^{\mathrm{C}}_{\mathrm{w}}$ or $H^{\mathrm{C}}_{\mathrm{w}}$, for two images f and g, respectively. For the two images, we define the L_p-norm for cells as

$$D_{L_p} = \left(\int_{\Omega} \int_0^{2\pi} |H^{\mathrm{C}}_P(\theta, \boldsymbol{c}) - H^{\mathrm{C}}_Q(\theta, \boldsymbol{c})|^p d\theta d\boldsymbol{c} \right)^{1/p}. \tag{5.42}$$

This L_p-norm discriminates the difference between all the cells dividing images. Furthermore, we define the L_p-norm for a collection of blockwise cells as

$$D_{L_p} = \left(\int_{\Omega} \int_{B(\boldsymbol{c})} \int_0^{2\pi} |H^{\mathrm{C}}_P(\theta, \boldsymbol{c}) - H^{\mathrm{C}}_Q(\theta, \boldsymbol{c})|^p d\theta d\boldsymbol{x} d\boldsymbol{c} \right)^{1/p}. \tag{5.43}$$

This L_p-norm discriminates the difference between all the cells obtained by moving a block. Moreover, we define the L_p-norm for blocks by setting $P(\boldsymbol{c}) = \{H^{\mathrm{B}}_P(\theta, \boldsymbol{c}) \mid 0 \leq \theta < 2\pi\}$ and $Q(\boldsymbol{c}) = \{H^{\mathrm{B}}_Q(\theta, \boldsymbol{c}) \mid 0 \leq \theta < 2\pi\}$, which are given by the histograms H^{B}, $H^{\mathrm{B}}_{\mathrm{w}}$ or $H^{\mathrm{B}}_{\mathrm{w}}$ for two images f and g, respectively. Then, we define the L_p-norm for a collection of blocks as

$$D_{L_p} = \left(\int_{\Omega} \int_0^{2\pi} |H^{\mathrm{B}}_P(\theta, \boldsymbol{c}) - H^{\mathrm{B}}_Q(\theta, \boldsymbol{c})|^p d\theta d\boldsymbol{c} \right)^{1/p}. \tag{5.44}$$

5.4.2 Global Directional Distribution Method

Let an image f be an image blurred by Gaussian filtering with standard deviation σ. For this blurred image f, using the histogram defined in eq. (5.27), we construct the global DD feature as

$$F^{\mathrm{G}}_{\mathrm{DD}}(f) = P = \{H^{\mathrm{G}}_{\mathrm{w}}(\theta) | 0 \leq \theta < 2\pi\}. \tag{5.45}$$

This feature represents the DD over a region of an image. Furthermore, using the histogram defined in eq. (5.33), we construct the global DDD feature as

$$F^{\mathrm{G}}_{\mathrm{DDD}}(f) = P = \{H^{\mathrm{G}}_{\mathrm{D}}(\theta) | 0 \leq \theta < 2\pi\}. \tag{5.46}$$

Both features consist of a histogram of the global DD over an image. To distinguish them from the features based on histograms of local DDs, we use an upper subscript G for the features based on a histogram of the global DD.

For images f and g, we extract the feature $F^{\mathrm{G}} \in \{F^{\mathrm{G}}_{\mathrm{DD}}, F^{\mathrm{G}}_{\mathrm{DDD}}\}$ as $F^{\mathrm{G}}(f) = P$ and $F^{\mathrm{G}}(g) = Q$. To discriminate the difference between two features, we

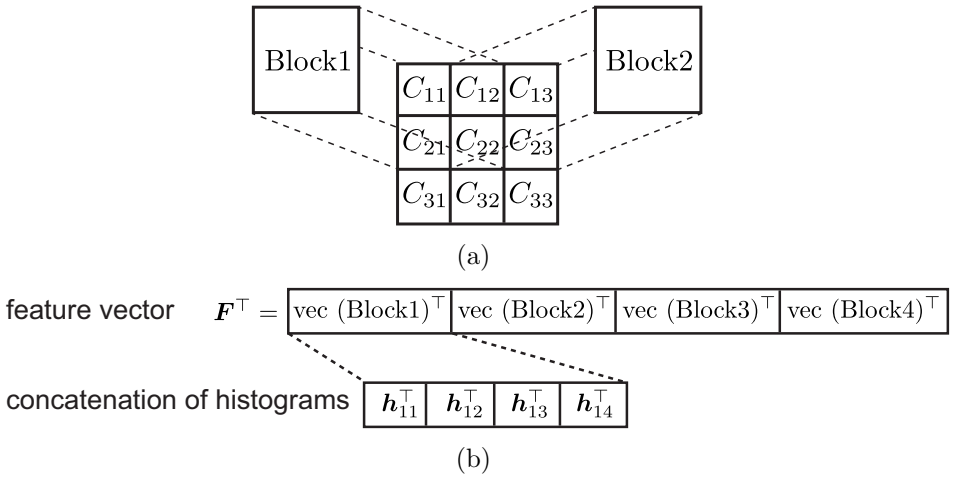


Figure 5.6: Feature extraction of the HoG method. As shown in (a), an image is divided into cells C_{ij} and the DDs of each local region $B_k, k \in \{1, 2, 3, 4\}$ are obtained by a moving window for $i, j \in \{1, 2, 3\}$. (b) shows how to construct a feature vector in the HOG method. Extracted histograms in each cell C_{ij} are represented as column vectors $\boldsymbol{h}_{i'j'}$. The vectorised histograms in each block are connected and normalised by the ℓ_2-norm. By connecting these ℓ_2-normalised vectors, we obtain the feature vector.

define the global discrimination method (**G method**), which discriminates a pair of image patterns using

$$D_{\mathrm{G}} = D_{\mathrm{W}}(P, Q), \tag{5.47}$$

where we use the Wasserstein distance. Furthermore, we define the L_p-norm between two features. Setting $P = \{H_P^{\mathrm{G}}(\theta) | 0 \leq \theta < 2\pi\}$ and $Q = \{H_Q^{\mathrm{G}}(\theta) | 0 \leq \theta < 2\pi\}$, we have

$$D_{L_p} = \left(\int_0^{2\pi} |H_P^{\mathrm{G}}(\theta) - H_Q^{\mathrm{G}}(\theta)|^p d\theta \right)^{1/p}, \tag{5.48}$$

where both histograms H_P^{G} and H_Q^{G} are either $H_{\mathrm{w}}^{\mathrm{G}}$ or $H_{\mathrm{D}}^{\mathrm{G}}$.

5.4.3 HOG-based Discrimination

The feature extraction in the HOG method consists of three steps.

1. Compute $\Phi(f)$ in section 3.2.
2. Compute the histogram of gradients defined in eq. (5.25) in a local region $C(\boldsymbol{c})$.

3. Extract the HoG feature of an image by sliding a bounding box.

The HOG method computes histograms for only quantised directions, here called oriented directions. We use L_p-normalisation for the histogram in each block as

$$H_{\mathrm{H}_p}(\theta, \boldsymbol{c}) = \frac{H_{\mathrm{w}}^{\mathrm{C}}(\theta, \boldsymbol{c})}{\left(\int_{\boldsymbol{x}\in B(\boldsymbol{c})} \int_0^{2\pi} |H_{\mathrm{w}}^{\mathrm{C}}(\theta, \boldsymbol{c})|^2 d\theta d\boldsymbol{x}\right)^{1/p}}. \tag{5.49}$$

In ref. [41], $H_{\mathrm{H}} = H_{\mathrm{H}_2}$ is used for the HOG method. Sliding the centre $\boldsymbol{c}$ of block $B(\boldsymbol{c})$ over region Ω of an image f, we extract the HOG feature as

$$F_{\mathrm{H}}(f) = \{H_{\mathrm{H}}(\theta, \boldsymbol{c}) \mid \boldsymbol{c} \in \Omega,\ 0 \leq \theta < 2\pi\}. \tag{5.50}$$

For two HOG features $F_{\mathrm{H}}(f) = \{H_{\mathrm{H},1}(\theta, \boldsymbol{c})\}$ and $F_{\mathrm{H}}(g) = \{H_{\mathrm{H},2}(\theta, \boldsymbol{c})\}$ of images f and g, respectively, L_p-norm

$$D_{L_p} = \left(\int_\Omega \int_0^{2\pi} |H_{\mathrm{H},1}(\theta, \boldsymbol{c}) - H_{\mathrm{H},2}(\theta, \boldsymbol{c})|^p d\theta d\boldsymbol{c}\right)^{1/p}. \tag{5.51}$$

is used. Discrimination of the difference between two HoG features using the criterion D_{L_p} is the HOG method in ref. [41].

We rederive L_p-norm for histogram of gradients. For a center point $\boldsymbol{y} = (y_1, y_2)^\top$ of a window function $W(\boldsymbol{y})$, histogram of directional gradients is a transformed $\nabla f(\boldsymbol{x})$ by a transformation T. Therefore, from eqs. (5.9) and (5.10) , we have

$$\|f\|_{p,T} = \sqrt[p]{\int_{R^2} w(\boldsymbol{x} - \boldsymbol{y}) T(|\nabla f(\boldsymbol{x})|^p) d\boldsymbol{x} d\boldsymbol{y}}, \tag{5.52}$$

where T nonlinearly transform local region of an image at each point. Using a window function, we define the transformed gradients $T(\nabla f)$ as

$$F(\theta, \boldsymbol{y}) = \frac{1}{|W|} \int_W |\nabla f(r\cos\theta - y_1, r\sin\theta - y_2)|^p r dr d\theta, \tag{5.53}$$

where $\boldsymbol{y} = (y_1, y_2)^\top$ is a center point of a window function. For local directional distributions on windowed regions, we have L_p-norm

$$\|f\|_{p,F} = \sqrt[p]{\int_{R^2} w(\boldsymbol{x} - \boldsymbol{y}) \frac{1}{|W|} \int_W |\nabla f(r\cos\theta - y_1, r\sin\theta - y_2)|^p r dr d\theta d\boldsymbol{x}}. \tag{5.54}$$

If we define transformed $\nabla f(\boldsymbol{x})$ as

$$F(\theta, \boldsymbol{y}) = \frac{1}{|W|} \int_W \frac{1}{\boldsymbol{x}} |\nabla f(\boldsymbol{x} - \boldsymbol{y})|^p d\boldsymbol{x}, \tag{5.55}$$

we have its polar coordinate representation as

$$F(\theta, \boldsymbol{y}) = \frac{1}{|W|} \int_W |\nabla f(r\cos\theta - y_1, r\sin\theta - y_2)|^p dr d\theta. \tag{5.56}$$

Then we have

$$\|f\|_{p,F} = \sqrt[p]{\int_{R^2} w(\boldsymbol{x} - \boldsymbol{y}) \frac{1}{|W|} \int_W |\nabla f(r\cos\theta - y_1, r\sin\theta - y_2)|^p dr d\theta d\boldsymbol{x}}. \tag{5.57}$$

If we use this norm for computation of distance of two images, discrimination by this norm is equivalent to the L_p-norm defined in eq. (5.51).

For block-based discrimination method for HOG features, setting $H_1^{'}(\boldsymbol{c}) = \{\int_{\boldsymbol{x}\in B(\boldsymbol{c})} H_{\mathrm{H},1}(\theta, \boldsymbol{x}) d\boldsymbol{x} \,|\, 0 \leq \theta < 2\pi\}$ and $H_2^{'}(\boldsymbol{c}) = \{\int_{\boldsymbol{x}\in B(\boldsymbol{c})} H_{\mathrm{H},2}(\theta, \boldsymbol{x}) d\boldsymbol{x} \,|\, 0 \leq \theta < 2\pi\}$, we define

$$D_{\mathrm{Block}} = \int_{\boldsymbol{c}\in\Omega} WD\left(H_1^{'}(\boldsymbol{c}), H_2^{'}(\boldsymbol{c})\right) d\boldsymbol{c}. \tag{5.58}$$

In this method, the difference between the DD and HoG features is how each histogram of a block is normalised. The HoG method defines the histogram of a block as connected vectorised histograms of cells normalised by the L_2-norm.

For the normalised histograms $H_{\mathrm{H}_p}(\theta, \boldsymbol{c})$ with $p \in \{1, 2\}$, we define the L_1-norm of a block in an image as $D_{L_{1,p}}(\boldsymbol{c}) = \int_0^{2\pi} |H_{\mathrm{H}_p}(\theta, \boldsymbol{c})| d\theta$. Therefore, we rewrite the L_1-norm defined by eq. (5.51) for the HoG feature as

$$D_{L_{1,p}} = \int_{\boldsymbol{c}\in\Omega} |D_{L_{1,p}}(\boldsymbol{c})| d\boldsymbol{c}, \tag{5.59}$$

with $p \in \{1, 2\}$. From Theorem 2, the inequality

$$D_{L_{1,1}} \leq \sqrt{n} D_{\mathrm{L}_{1,2}} \tag{5.60}$$

holds. Here, n is the number of bins of histograms in a block. The connection of four histograms gives a larger upper bound of the L_1-norm than that given by only one histogram. Furthermore, from Theorem 3, we can infer that L_2-normalisation makes the separation ratio higher than that for L_1-normalisation in the recognition of HoG features.

5.5 Experiments

By analysing the performances of the recognition rate for local and global directional-distribution-based methods and the HoG method, we examine the important mathematical properties required for gradient-based image pattern recognition. To evaluate the performances of these features, we compute the recognition rate and receiver operating characteristic (ROC) curve.

For feature extraction, adopting the same procedure as in the original HoG paper [41], we use 9 and 18 oriented directions given by $\{\frac{\pi}{9}(i-1)\}_{i=1}^{9}$ and $\{\frac{\pi}{9}(i-1)\}_{i=1}^{18}$, respectively. The experimental results in ref. [41] show that there is no significant difference between the selection of 9 and 18 directions for the HoG feature. For local and global features, we use 9 and 18 directions, respectively. For the HoG feature, we use both 9 and 18 directions. Using these quantised directions, we derive the discretised direction of the gradient as $\angle_d(\nabla f) = \theta^* = \arg\max_{\theta_i}\left(\frac{\boldsymbol{\omega}(\theta_i)^\top \nabla f}{\|\nabla f\|_2}\right)$, where $\boldsymbol{\omega}(\theta_i) = (\cos\theta_i, \sin\theta_i)^\top$. For this discretised direction θ^*, we define the magnitude of the gradient as $m_{\mathrm{d}}(\boldsymbol{x}) = \|\boldsymbol{\omega}(\theta^*)^\top \nabla f(\boldsymbol{x})\|_2$. Therefore, we use a pair of a discretised direction and its magnitude $\langle \theta^*, m_d(\boldsymbol{x})\rangle$ to construct a histogram of the DD. The sizes of a cell and block are 8×8 pixels and 2×2 cells, respectively. For the DDD, we set $\lambda = 0.9$.

Throughout this section, as metrics, we use the L_1- and L_2-norms, the p-Wasserstein distance with $p = 1$ (**1WD**) and the binomial-distribution-based p-Wasserstein distance with $p = 1$ (**B1WD**). For the local SDD, DD and DDD features, we use the C method, B method and BWC method and the L_p-norms with $p = 1, 2$ defined in eqs. (5.39), (5.41), (5.40), and eqs. (5.42), (5.44) and (5.43), respectively. For the global DD and DDD features, we use the G method and the L_p-norm defined in eqs. (5.47) and (5.48), respectively. For the HoG feature, we use the B method and the L_p-norm with $p = 1, 2$ defined in eqs. (5.58) and (5.51), respectively. Table 5.3 summarises these settings for each pair of a feature and a discrimination method.

For the computation of the recognition ratio and ROC curve, we use the INRIA dataset [41]. The INRIA dataset contains high-quality annotations of pedestrians in diverse settings (cities, beaches, mountains, etc.). Since the dataset is widely used for the performance evaluation of detection, it is suitable for our validation.

From the INRIA dataset, we select images that show frontal views of pedestrians as positive images. These positive images are divided into 38 learning positive images and 115 positive queries. To obtain negative images, we randomly crop 115 background regions of images in the INRIA dataset,

Table 5.4: Details of computation of discrimination method for aggregated DD. '♯Histogram' represents the number of histograms that are used in the discrimination. '♯Direction' represents the number of quantised directions. 'Computational time' shows the average computational time of each discrimination method given by the Wasserstein distance (WD) and L_p-norm (L_p). For practical computation, we use a Xeon X5570 2.93GHz (quad core) processor. The HoG method is included in the types of blocks. 'No division' represents the global method.

Division type	♯Histogram	♯Directions	Dimension of vectorised feature	Computational time (WD)	Computational time (L_p)
Cells	16×8	9	1152	1.5539 [sec]	0.0001 [sec]
		18	2304	2.4640 [sec]	0.0001 [sec]
Blocks	15×7	9	945	1.2747 [sec]	0.0001 [sec]
		18	1890	2.0213 [sec]	0.0001 [sec]
Blockwise cells	$15 \times 7 \times 4$	9	3780	5.0988 [sec]	0.0001 [sec]
		18	7560	8.0850 [sec]	0.0001 [sec]
No division	1	9	9	0.0121 [sec]	–
		18	18	0.0145 [sec]	–

which we use as negative queries. Figures 5.7 (a), (b) and (c) show positive learning images, examples of positive queries and examples of negative queries, respectively. The size of all images is 130×70 pixels. For extraction, we use the centre 124×64 region of images. Table 5.4 gives details of the parameters and computational times for the recognition.

For all the combinations of the feature and the discrimination method, we compute the median of the learning images as a preliminary step. We compute the distances from each median to all queries using the same feature extraction and discrimination methods for each median. Using the computed distances, we classify queries with a criterion. We define a threshold as $X\%$ of the largest distance between the median and query, where we set $X \in \{5, 10, 15, \ldots, 100\}$. If a distance is less than or equal to the threshold, we classify a query as a positive image and vice versa. This threshold defines the space of positive queries from learning data. If all the queries exist in a small space defined by a small threshold, we achieve robust recognition since the false positive rate is small in the recognition. If a small threshold gives the highest recognition rate, the results mean that a combination of a feature and a discrimination method achieves discriminative classification. Algorithms 5.1 and 5.2 summarise the classification procedure for a pair of a feature and a discrimination method.

Figure 5.8 shows the recognition rates and ROC curves for the local SDD and DD features and the HoG feature. Figures 5.9 and 5.10 show the recogni-

(a) Positive images for deciding median

(b) Examples of positive queries

(c) Examples of negative queries

Figure 5.7: (a) Set of 38 positive images for deciding a median. (b) Examples of positive queries of pedestrians. The set of positive queries does not include the set of 38 images in (a). The total number of positive queries is 115. (c) Examples of negative queries of pedestrians. The total number of negative queries is 115. All images have a resolution of 130×70 pixels. For feature extraction, we use only the centre region of 124×64 pixels of these images.

Algorithm 5.1: Preliminaries for classification

Input: N_{L} learning data $\{f_k\}_{k=1}^{N_{\mathrm{L}}}$.

Output: The median of the learning data f_{M}.

1. Extract feature F_k of image f_k for all learning data.
2. Compute distance $D(F_i, F_j)$ for all pairs of F_i and F_j with $i, j \in \{1, 2, \ldots, N_{\mathrm{L}}\}$.
3. Find the median f_{M} such that its extracted feature $F_{\mathrm{M}} = \arg\min_{F_i} \left(\sum_j^{N_{\mathrm{L}}} D(F_i, F_j) \right)$.

Algorithm 5.2: Classification of queries

Input: N queries $\{f_i\}_{i=1}^{N}$, an extract feature F_{M} of the median for distance $D(\cdot, \cdot)$, threshold X.

Output: Labels $\{l_i\}_{i=1}^{N}$, which are positive (1) or negative (-1) for each query.

1. Extract features $\{F_i\}_{i=1}^{N}$ from all queries $\{f_i\}_{i=1}^{N}$.
2. Compute distance $D(F_{\mathrm{M}}, F_i)$ between the median and query for all queries.
3. Decide labels for each query as
$$l_i = \begin{cases} 1, & \text{if} \quad D(F_{\mathrm{M}}, F_i) \leq \frac{X}{100} D_{\max} \\ -1, & \text{otherwise,} \end{cases}$$
where $D_{\max}$ is the largest distance between the median and query, for $i = 1, 2, \ldots, N$.

tion rates and ROC curves for the global DD and DDD features, respectively. Figure 5.11 shows the recognition rates and ROC curves for the local DDD features.

In Figs. 5.8(a) and (c), recognition using the L_1-norm for blocks of local directional features achieves a recognition rate higher than 0.5. As shown in Figs. 5.8(b) and (d), all the discrimination methods except the B method using the L_1-norm are not discriminative for local directional distribution features. As shown in Figs. 5.8(a)-(d), the local DD feature gives a larger recognition rate than the local SDD feature for all discrimination methods. Figure 5.8(e) shows that discrimination using the L_1-norm gives the highest recognition rate for the HoG feature with both 9 and 18 directions. Figures 5.8(a), (c) and (e) show that discrimination using the HoG feature gives a higher recognition rate than that using the local SDD and DD. Furthermore, Fig. 5.8(e) shows that discrimination using the L_2-norm gives the smallest recognition rate for both 9 and 18 directions.

Comparing with Fig. 5.8, Fig. 5.9 shows that the G method has a higher

Table 5.5: Summary of discriminative combinations of features and metrics.

Extraction method	Local method		Global method	
	DD feature	HoG feature	DD feature	DDD feature
Discriminative metric	L_1	L_1	L_1 and L_2	1WD

recognition rate than the C method, B method and BWC method. Figures 5.9(e) and (f) show that the discrimination of blurred images by Gaussian filtering with a larger standard deviation gives a more discriminative feature for the global DD method.

In Fig. 5.10, the G method with 1WD possesses a higher recognition rate than that with the L_1- and L_2-norms and B1WD. These results show that if we use a DDD, we have the same recognition rate as for the HoG method with 9 oriented directions and the L_2-norm shown in Fig. 5.8(e). In Fig. 5.11, discrimination by the C method and BWC method with 1WD gives the highest recognition rate.

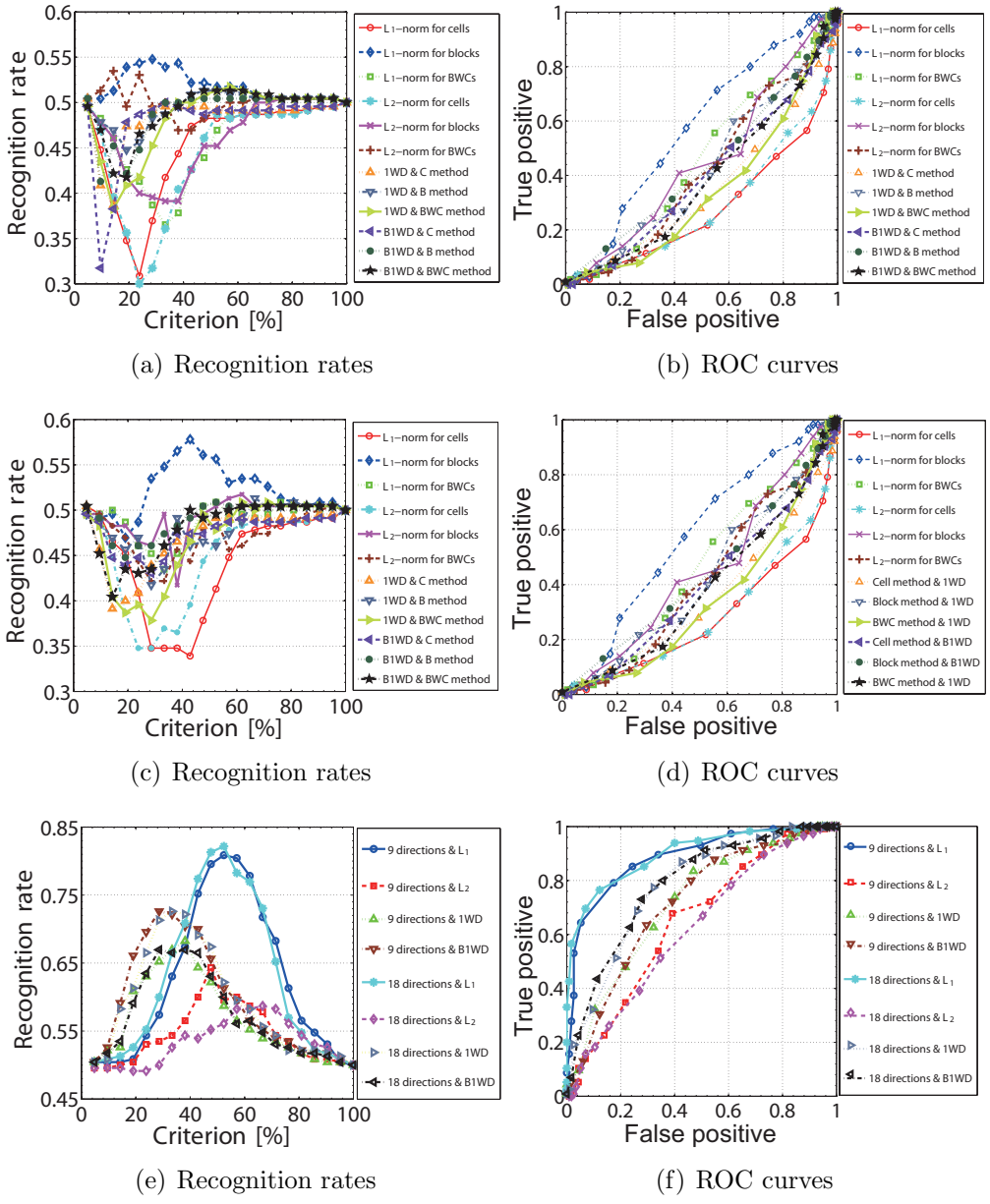
Figure 5.12 shows the relation between the L_1- and L_2-normalisations of histograms in the HoG feature. Figure 5.12(a) shows that the L_2-normalisation of histograms in each block in the HoG feature has a larger distance between the median and queries than the L_1-normalisation of histograms in each block in the HoG feature. The L_1-norm of the HoG feature whose histograms are normalised by the L_2-norm is bounded by Theorem 3. In the HoG method, by connecting four histograms in a block, this upper bound becomes larger than that for one histogram. In Fig. 5.12(a), the nonlinearity between the distances of features given by L_1- and L_2-normalisations for histograms in the HoG feature is also illustrated. In the Figs. 5.12(b) and (c), results show the distributions of distances given by the L_1- and L_2-normalisations of histograms in the HoG feature, respectively.

Table 5.5 summarises the discriminative combinations of features and metrics. In the context of directional statistics, the combination of the Wasserstein distance and the DDD feature gives accurate recognition. Using the L_1-norm for the DD feature, we obtain the same performance without the extraction of the dominant directions.

5.6 Discussion

For the results of the previous section, we summarise four key observations about feature extraction methods and discrimination methods.

From Figs. 5.8(a) and (b), and Fig. 5.9 the first observation is that

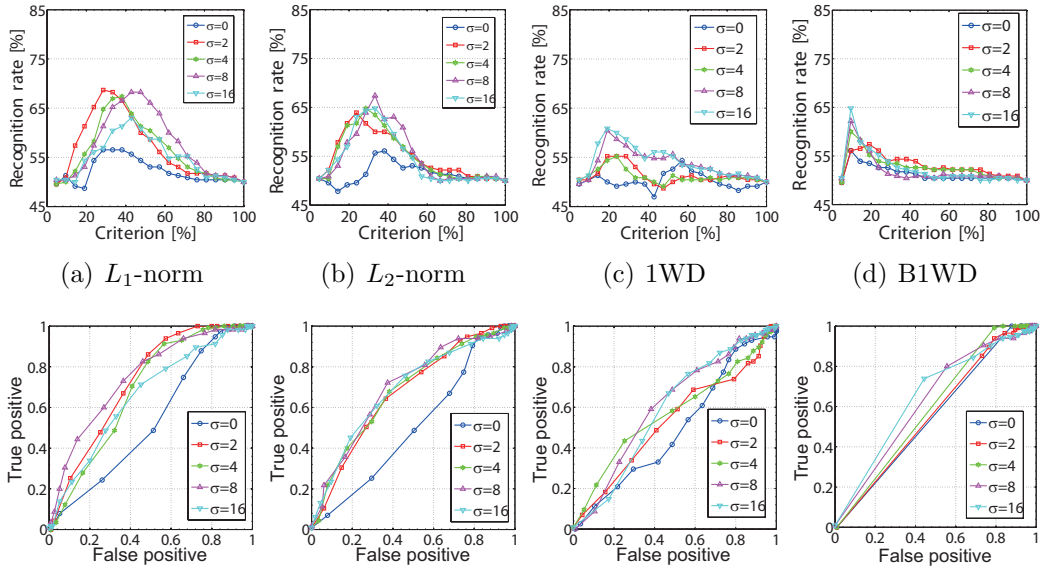


(a) Recognition rates (b) ROC curves

(c) Recognition rates (d) ROC curves

(e) Recognition rates (f) ROC curves

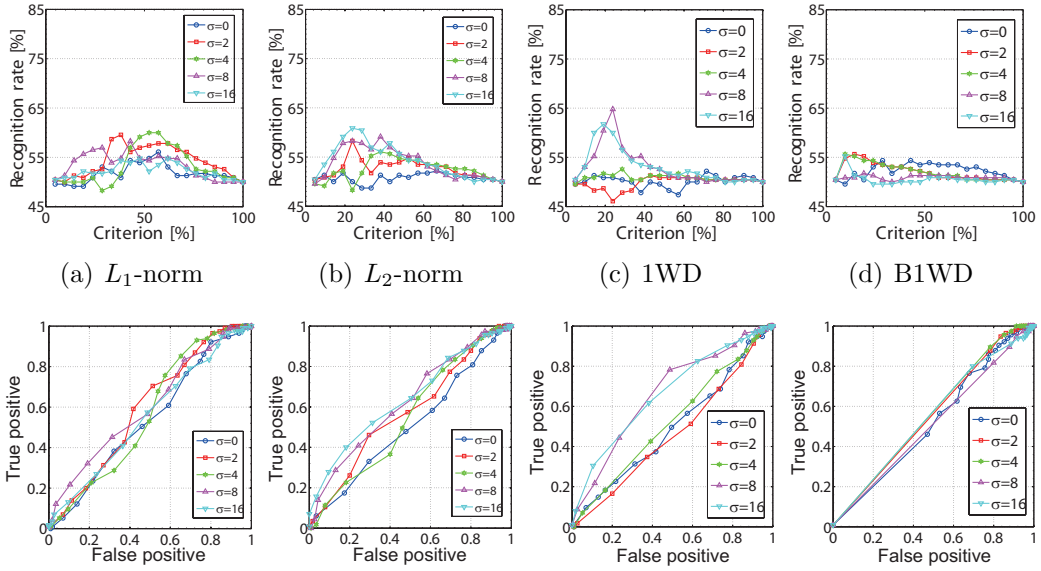
Figure 5.8: Recognition rates and ROC curves for the simple directional distribution (SDD) feature, directional distribution (DD) feature and histogram of oriented gradients (HoG) feature. Left and right columns show the results for the recognition rate and ROC curve, respectively. Top, middle and bottom rows show results for the SDD, DD and HoG features, respectively. In (a), (c) and (e), the vertical and horizontal axes represent the recognition rate and criterion, respectively. In (b) (d) and (f), the vertical and horizontal axes represent the true positive rate and false positive rate for each given criterion, respectively. The discrimination method using the L_1-norm gives the highest recognition for the SSD, DD and HoG features.



(a) L_1-norm (b) L_2-norm (c) 1WD (d) B1WD

(e) L_1-norm (f) L_2-norm (g) 1WD (h) B1WD

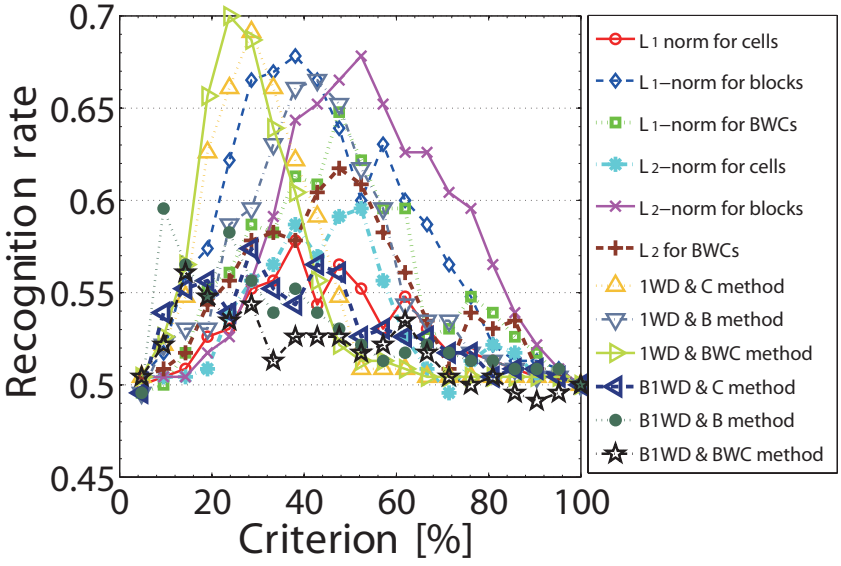
Figure 5.9: Recognition rates and ROC curves for the global DD features. Upper and lower rows respectively show the recognition rates and ROC curves. The first, second, third and fourth columns show the results for discrimination using the L_1-norm, L_2-norm, 1-Wasserstein distance (1WD) and binomial-distribution-based 1-Wasserstein distance (B1WD), respectively. In (a)-(d), the vertical and horizontal axes represent the recognition rate and criterion, respectively. In (e)-(h), the vertical and horizontal axes represent the true positive rate and false positive rate, respectively. In (a)-(d), circles, squares, six-rayed stars, and upward and downward triangles represent results for blurred images with Gaussian filtering with standard deviations of $0, 2, 4, 8$ and 16, respectively.



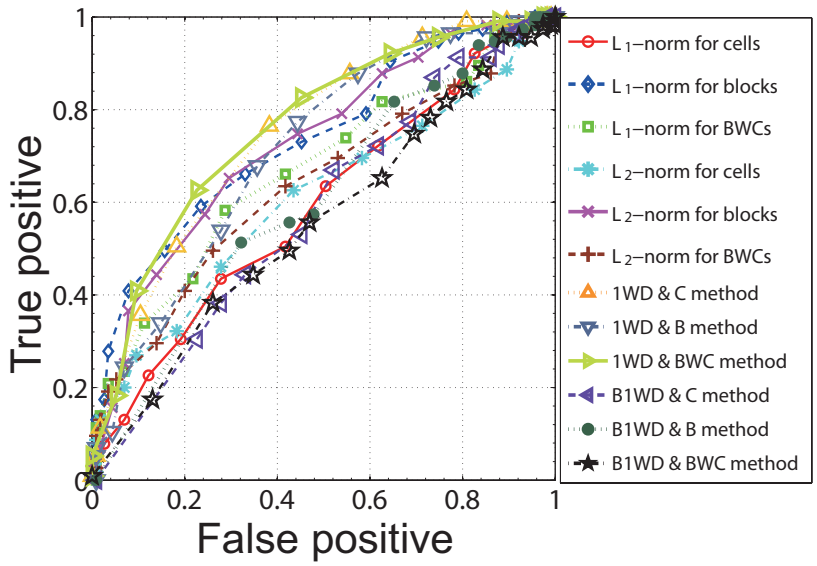
(a) L_1-norm (b) L_2-norm (c) 1WD (d) B1WD

(e) L_1-norm (f) L_2-norm (g) 1WD (h) B1WD

Figure 5.10: Recognition rates and ROC curves for the global DDD features. Upper and lower rows respectively show the recognition rates and ROC curves. The first, second, third and fourth columns show the results for the discrimination using the L_1-norm, L_2-norm, 1-Wasserstein distance (1WD) and binomial-distribution-based 1-Wasserstein distance (B1WD), respectively. In (a)-(d), the vertical and horizontal axes represent the recognition rate and criterion, respectively. In (e)-(h), the vertical and horizontal axes represent the true positive rate and false positive rate, respectively. In (a)-(d), circles, squares, six-rayed stars, and upward and downward triangles represent the results for blurred images with Gaussian filtering with standard deviations of $0, 2, 4, 8$ and 16, respectively.



(a) Recognition rates



(b) ROC curves

Figure 5.11: Recognition rates and ROC curves for the DDD features. In (a), the vertical and horizontal axes represent the recognition rates and criteria, respectively. In (b), the vertical and horizontal axes represent the true positive rate and false positive rate for each given criterion, respectively.

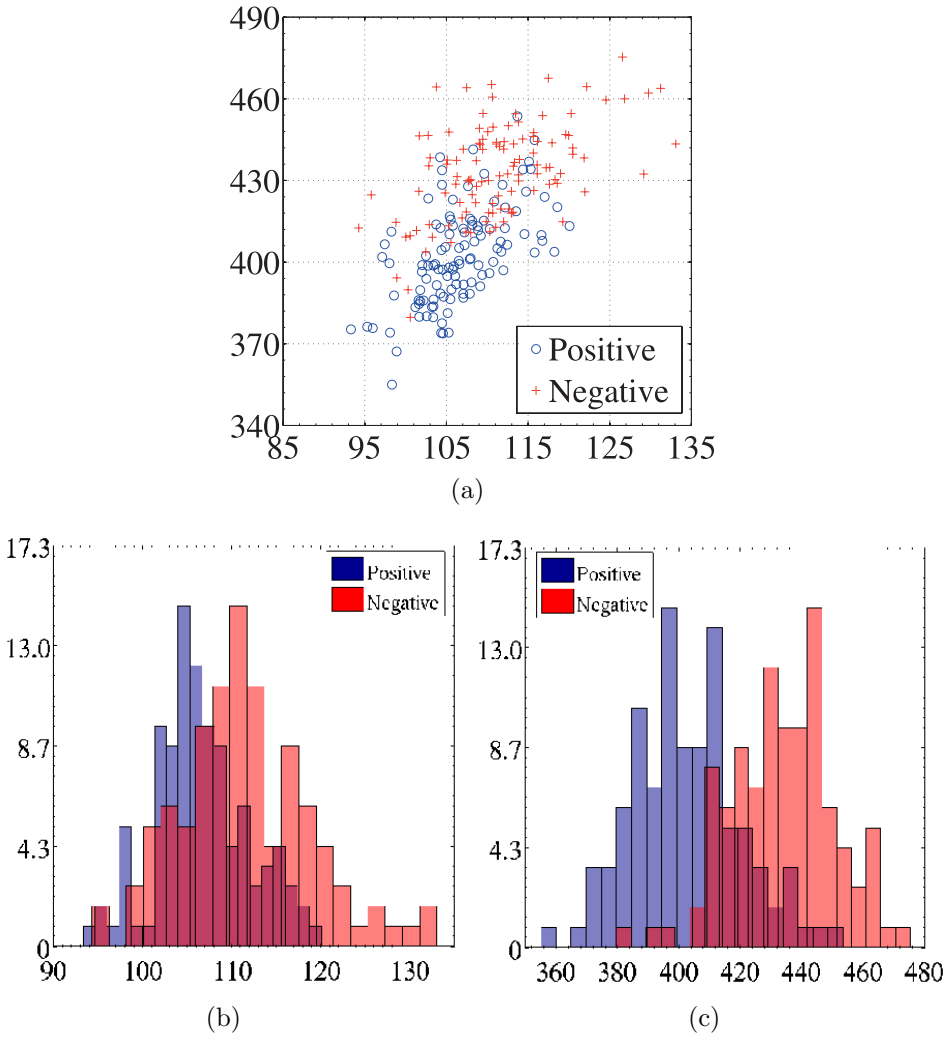


Figure 5.12: Distribution of L_1-norms among the median and queries. (a) Relation between L_1-norms for L_1- and L_2-normalised HoG features. The horizontal and vertical axes represent L_1-norms for L_1- and L_2-normalised HoG features, respectively. This relation shows that the mapping ϕ is nonlinear. In (b) and (c), the horizontal and vertical axes represent the distances between the median and the queries and their probability of occurrence, respectively. L_2-normalisation gives more discriminative distributions for positive and negative queries than L_1-normalisation.

the low-frequency features of an image are important for the image pattern recognition. An image can be represented as a linear combination of functions such as low-frequency and high-frequency sinusoidal functions. The frequency of the features depends on the size of local regions. If the regions are large, the features with a low frequency are extracted. Therefore, the histograms are constructed with respect to block-extracted features with a low frequency since the blocks are larger than the cells and blockwise cells. Obviously, the recognition ratio of the block-extracted features in Figs. 5.8 (a) and (c) is higher than cell- and blockwise cell-extracted features. The results of Fig. 5.9 also imply that the discriminative features of distributions of gradients depend on the low frequency components of an image since Gaussian filtering extracts the features of low-frequency components of an image.

The comparison between Figs. 5.8 and 5.9 implies that the the aggregation of the local DDs has a similar role to the Gaussian filtering as the second observation. The shape of the graphs of the recognition rate in Figs. 5.9(a)-(d) is similar to that of the graphs in Fig. 5.8(e), which are given by the 1WD and B1WD. This comparison shows that discriminating differences between DDs in local regions can be approximated by the extraction of DDs in blurred global regions. For aggregated local regions, the L_1- and L_2-norms have a similar role of blur-filtering an image since the discrimination using the L_1- and L_2-norms for the SSD and DD involves summing all the differences between local regions. Compared with the method of aggregating the DDs in local regions, the G method has a smaller data size that is 0.2% of the size of the HoG feature with 18 directions as shown in Table 5.4. Therefore, the G method is a acceptable approximation of the HoG method. Furthermore, the results in Fig. 5.9 imply that aggregating local regions into a block is not important for gradient-based methods since the G method possesses higher recognition rate than C method, B method and BWC method.

The third observation from Figs. 5.8(a)-(d) is that discriminative DDs represent edges in an images. The results in Figs. 5.8(a)-(d) show that the magnitudes of gradients are essential for image pattern recognition, since the distribution of the magnitudes of gradients determines the edge shape. Figure 5.10 also implies the importance of the edges in an images since this result show the dominant direction for gradient-based image pattern recognition. The discrimination using B1WD and the L_1-norm gives a lower recognition rate than that with the 1WD since the gap between indices in the histogram is large and the value of the histogram is almost the same as that in the DDD. Comparing Figs. 5.8, 5.9, 5.10 and 5.11, the local DDD feature with discrimination using 1WD gives the highest recognition rate among the DD-based methods. These results show that if we use the DDD, we have the

same recognition rate as the HoG method with 9 oriented directions and the L_2-norm.

The fourth observation is for discriminative properties of the gradient histogram. The results for the HoG feature in Fig. 5.8(e) imply that the HoG feature is not a probabilistic distribution of gradients but a feature based on gradients, since the results are different from the results of B method in Figs 5.8(c). The difference between the B method and HoG method is the normalisation method. We presented the properties of the normalisation of the HoG method as Theorems 2 and 3. Figure 5.12 shows that the nonlinear mapping from the L_1-normalised HoG feature to the L_2-normalised HoG feature gives a more discriminative distribution of the distance.

The results in our experiments imply that measuring the difference between aggregated distributions of gradients in local regions of images is not a discriminative method in the context of the image pattern recognition. For the construction method of histograms, both the magnitudes and directions of gradients are important since a pair of them can represent the smoothness and the directions of gradients.

If we use the L_1-norm for the histogram of oriented gradients features, we obtain a higher recognition rate than using the L_2-norm and the Wasserstein distances. This results is coincident to eq.(5.60) We show that the global method is a discriminative feature compared with the methods based on the directional distributions in local regions. Furthermore, the results of a comparison between the histogram of oriented gradients method and the global method imply that the histogram of oriented gradients method extracts the low-frequency features of local regions. If we use L_1- and L_2-norms, aggregating has a similar role to blurring for images. Moreover, the dominant directional distribution for image pattern recognition is the discriminative directional distribution. If we use the Wasserstein distance for the dominant directional distribution features, we obtain the highest recognition rate among the directional-distribution-based methods. These dominant directional distributions represent edges such as the structure lines [50] in images. Using the L_1-norm for the directional distribution in both the local and global directional distribution methods, we achieve accurate detection without the extraction of dominant directions.

5.7 Summary

We introduced metrics for directional distributions of gradients for image pattern recognition. Firstly, we defined three methods for constructing histograms of directional distributions. The first method is based on only the

directions of gradients. The second method is based on both the directions and magnitudes of gradients. The third method is based on the dominant directions of gradients. Secondly, we defined a dividing method for images based on cells and blocks. On the basis of these divided regions, we defined the simple directional distribution feature, the directional distribution feature and the dominant directional distribution feature. For matching, we defined the cell method, block method, blockwise cell method, global method and L_p-norm to measure the difference between the directional distributions. Furthermore, we gave a functional representation and functional analysis for the histogram of oriented gradients method. Finally, in experiments using the cell method, block method, blockwise cell method and global method, we evaluated the performances of all the features.

From the results of our experiments, we have following observations. For the image pattern recognition, the discriminative feature is the edges of an image. In the context of the directional distribution, the dominant directional distributions represent the edges of a blurred image. For the image pattern recognition with the dominant directional distribution, the Wasserstein distance is an appropriate metric. As the acceptable approximation of the fast and accurate recognition with a pair of the dominant directional distributions and the Wasserstein distance, we can use the pair of the global directional distribution and L_1-norm. The comparisons between the histogram of oriented gradients method with our defined methods show that the feature of the histogram of oriented gradients method does not represent the probabilistic distribution of directions of gradients. By the normalisation with the L_2-norm, the feature of histogram of oriented gradients possesses more discriminative distribution in a feature space than the features that represent probabilistic distributions of gradients. These clarified mathematical properties are beneficial for those who use the histogram of oriented gradients method and who design a new feature extraction method based on the gradients of an image.

Chapter 6

Estimation of Geometrical Transform

This chapter is based on two published works. The following contents of this chapter for two- and three-dimensional images are based on Publications of International Conferences, "12. Two-Dimensional Global Image Registration Using Local Linear Property of Image Manifold" and "10. Global Volumetric Image Registration Using Local Linear Property of Image Manifold", respectively.

6.1 Manifold Learning for Image Registration

We propose a method of generating a new entries in a image dictionary from entries in a sparse dictionary. Using this generation method, we develop a global image registration method with a sparse dictionary.

In global image registration, the optimal transformation between template and reference images is accomplished by computing the best matching between these two images. Therefore, for accurate registration, we are required to prepare as many reference images as possible in an image dictionary of reference images. This implies that the spacial complexity of global image registration depends on the population of reference images in the dictionary. The generation of a new reference image from the existing reference images reduces the spacial complexity of global image registration. In image registration with a sparse dictionary that consists of a small number of reference images, we are required to generate a new entry and to compute the transform from this new entry simultaneously.

Since the image pattern space is a curved manifold in higher-dimensional space, on the tangent space of this curved manifold, an image pattern is

expanded to a finite Fourier series using local bases. This expansion means that local bases span a local part of the manifold. Figure 6.1(a) illustrates a manifold on a low-dimensional subspace [181, 36, 65]. Moreover, in the neighbourhood of an image pattern, image patterns can be expressed as a linear combination of this image and derivatives of this image. We call this property of an image manifold the local linear property. Figure 6.1(b) shows the generation of a template image. By combining these two local expressions for an image pattern, we obtain two benefits. One is that we can compute the parameters of the transform. The other is that we can compute a new pattern that is sufficiently close to a reference image. Figure 6.1(c) shows the relation between a generated image g^* and the nearest neighbour in a local subspace. In Fig. 6.1(c), the perturbation $\boldsymbol{\delta}_\theta$ is small because the generated g^* is close to the nearest neighbour. If the rotation angle, scaling factor and shear ratio are all small, these transforms can be expressed as a linear sum of the identity transform and linear transforms with parameters that define the type of transform. Therefore, we can decompose an affine matrix into rotation, scaling and shearing to estimate the parameters.

To use the local linear property for image registration, first, a curved manifold of image patterns is generated from reference images. For a template image, we generate this curved manifold using the k-neighbourhood method. To reduce the temporal and spacial complexities of the search of the k-neighbourhood, a dimension-reduction method is used for the generation of the curved manifold. For the dimension-reduction method, We adopt a random projection.

The random projection is a metric-embedding method that approximately preserves distances between points in the original space [166]. Furthermore, for an arbitrary set of points, the random projection preserves angles among points, the volumes of simplexes [118], the lengths of smooth curves [4] and manifolds [14]. Therefore, the random projection is used for approximation in the nearest-neighbour search [13]. Furthermore, the validity of the random projection for the dimension reduction of noiseless images, noisy images and text data is shown in [19]. In [19], experimental results show that the random projection preserves distances among the original high-dimensional data in a low-dimensional subspace. Therefore, we can use the random projection for manifold learning in a low-dimensional subspace.

Using the local linear property of a image manifold in the low-dimensional Euclidean space and the decomposition of affine transform, we have a linear equation system. This linear equation system represents relation between topological change on a manifold and geometrical change of images for a template and reference images. Therefore, solving this equation system, we can estimate parameters for the affine transform between the two images.

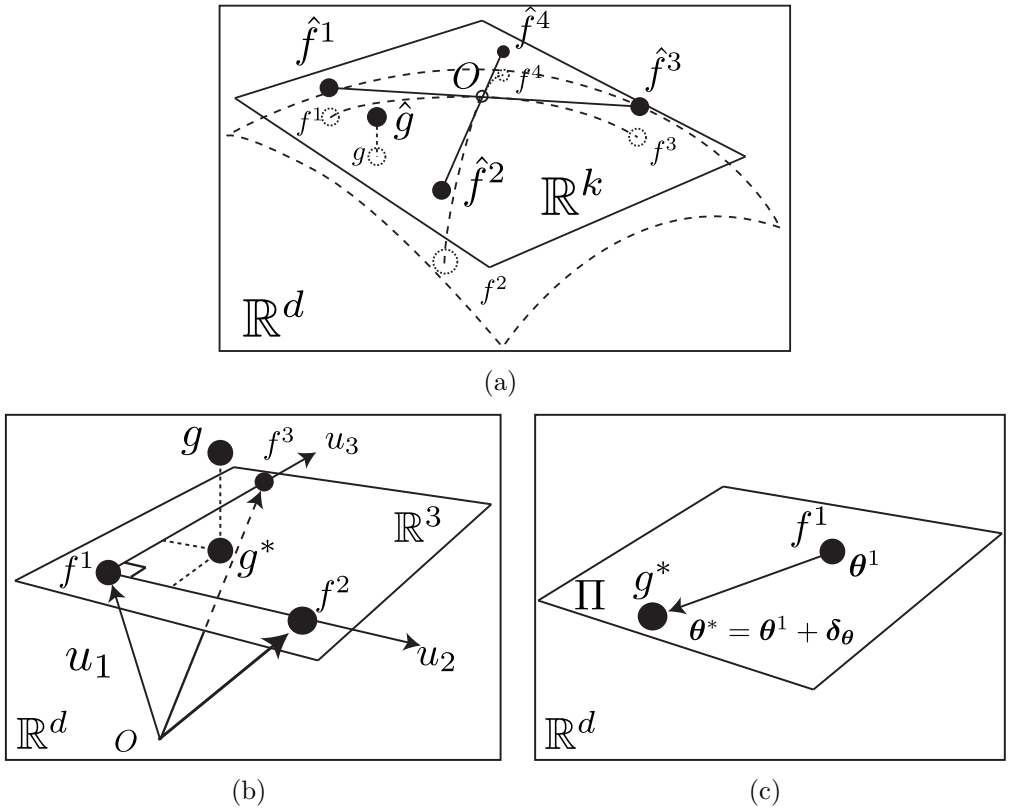


Figure 6.1: (a) Nearest neighbours of g searched for by the k-nearest-neighbour search on a manifold. Our method of projecting the manifold to a low-dimensional subspace. (b) Generation of a new entry in a dictionary. The input image g is projected onto the subspace spanned by three nearest neighbours. (c) Interpolation of parameter. For the new entry g^*, we interpolate the parameter θ^* of the image g^*. Here, Π represents the parameter space of the transform.

6.2 Related Works

For two-dimensional image registration in the computer vision and medical imaging, we can define three types of problems as shown in Fig. 6.2 In the first problem, a template image and a number of stored reference images are given. We are required to find the reference image that best matches the template image. After finding the best matching image, we estimate geometrical transform between the template and reference images. Second problem is to find a location of a template image in a reference image. To locate the template image within the reference image, we are require to find the best-match position of the template image in the reference image. In third problem, we are require to find the transform between a template image and a reference image that are taken by different cameras with different position and direction. Note that the second problem is a special case of the third problem.

In remote sensing [11, 102, 15] and computer vision [186, 7, 93], the second problem appear as a template matching. This template matching is important for localization in map and application for automatic factorisation. To solve the second problem, Kuglin and Hines adopt phase correlation of two images [97]. They proposed the phase-only correlation method that accurately estimate translation and rotation, that is, a rigid transform.

For the third problem, computer vision defines structure from motion (SfM) method [109, 69]. The SfM method reconstructs three-dimensional structure from estimation of relative motion of a camera using pinhole-camera model. To solve the third method, Torr *et al.* [159], and Pritchett and Zisserman [138] applied epipolar geometry and random sample consensus (RANSAC) to short- and wide-baseline matchings, respectively. Combination of epipolar geometry and the RANSAC that based on corresponding points in two images allow us to estimate homography between two images [70]. For further robust estimation of geometry, Torr and Zisserman proposed two modification of RANSAC called M-estimator sample and Consensus and maximum likelihood estimation sample and consensus [158]. Chum and Matas proposed randomised RANSAC to reduce time complexity of the RANSAC [34].

Although the above approaches mainly dealt with linear transform between images, medical image registration categorised into linear and nonlinear methods. For nonlinear image registration, global image registration is used as preprocess because nonlinear image registration is mainly valid for local deformation between images [162, 29, 80]. Nonlinear registration is used to detect optimally local transform between a template and the reference images, if the difference between these two images are small and local

[89]. Linear registration, however, detects global geometric relations between a template and the references [32].

Brown surveyed standard techniques [29] for two-dimensional registration. If a template image has only local differences from a reference image, nonlinear registration can be used to obtain the optimal local transform based on physical mode such that elastic, fluid and diffusion models [28, 33, 127, 27]. Therefore, for the accomplishment of nonlinear registration, medical image registration requires optimal linear registration.

In medical imaging, registrations are also used to overlap images that are captured by different instrument such that CT and MRI, CT and PET, and MRI and ultra sound data [154, 127, 136]. Furthermore, organs of each person have pattern perturbations even within same organ. Therefore, global registration in medical imaging adopt a priori knowledge as landmark given by medical expert and statistical information. Alpert *et al.* proposed principal axis transform for medical image registration [9]. Finding and overlapping principal axes between two images, we can establish global registration. Pluim *et al.* surveyed mutual information based image registration method. After maturing of the computer vision technique, medical imaging started to import these method. Dusty *et al.* [147], Sergey *et al.* [150] and Moradi *et al.* [126] adopt the SIFT for landmark in image registration.

6.3 Global Image Registration

Setting Π to be an appropriate parameter space for image generation, we assume that images are expressed as $f(\boldsymbol{x}, \boldsymbol{\theta}_i)$ for $\exists \boldsymbol{\theta}_i \in \Pi$, $\boldsymbol{x} \in \Omega$, $\Omega \in \{\mathbb{R}^2, \mathbb{R}^3\}$. The parameter $\boldsymbol{\theta}_i$ generates a transform for $f(\boldsymbol{x})$. We call the set of generated images $f(\boldsymbol{x}, \boldsymbol{\theta}_i)$ and parameters $\{\boldsymbol{\theta}_i\}_{i=1}^N$ a dictionary.

For the global alignment of images with respect to the region of interest Ω, we find the linear transformation $\boldsymbol{x}' = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{t}$ that minimises the criterion

$$R(f, g) = \sqrt{\int_\Omega |f(\boldsymbol{x}') - g(\boldsymbol{x})|^2 d\boldsymbol{x}} \tag{6.1}$$

for functions $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ defined on $\mathbb{R}^3$ such that

$$\int_\Omega |f(\boldsymbol{x})|^2 d\boldsymbol{x} < \infty, \quad \int_\Omega |g(\boldsymbol{x})|^2 d\boldsymbol{x} < \infty. \tag{6.2}$$

In image registration, we assume that the parameter $\boldsymbol{\theta}_i$ in Π generates the affine coefficients $\boldsymbol{A}$ and $\boldsymbol{t}$. Solving the NNS problem using the dictionary, we can estimate the transform $\boldsymbol{A}$ and $\boldsymbol{t}$ as $\boldsymbol{\theta}_i$. The computational cost of a

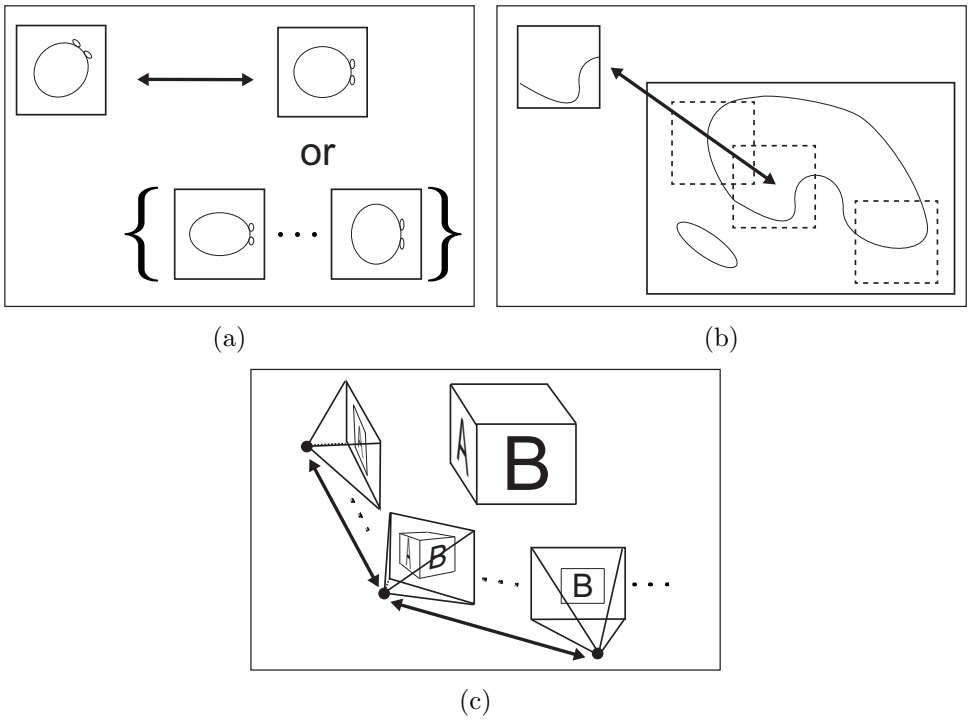


Figure 6.2: Registration problems in the computer vision. (a) Registration to one of the stored images. (b) Registration to the local region of an image. (c) Registration based on the camera model.

naive approach for the NNS is $\mathcal{O}(Nd)$, where N and d are the cardinality of the set of points in the metric space and the dimension of the metric space, respectively. The factor d in the nearest-neighbour search [166] is reduced by using the random projection. Furthermore, using the local linear property in section 6.4.2, we can also reduce N in the nearest NNS.

6.4 Local Eigenspace

6.4.1 Two-Dimensional Image

Setting the Hilbert space H to be the space of patterns, we assume that in H, the inner product (f, g) is defined. Furthermore, we define the Schatten product $\langle f, g \rangle$, which is an operator from H to H. Let $f \in H$ and P be a pattern and an operator for a class, respectively. We then define the class $\mathcal{C} = \{f \,|\, Pf = f, P^*P = I\}$. For recognition, we construct P for $f \in \mathcal{C}$ while minimising $E[\|f - Pf\|_2]$ with respect to $P^*P = I$, where $f \in \mathcal{C}$ is the

pattern for a class, I is the identity operator and E is the expectation in H. This methodology is known as the subspace method [76, 130, 176]. For the practical calculation of P, we adopt the Karhunen-Loeve expansion, which approximates the subspace of data in H.

We deal with images $f(\boldsymbol{x})$ defined on the two-dimensional Euclidean plane $\boldsymbol{x} = (x, y)^\top \in \mathbb{R}^2$. We assume that a small perturbation of the parameter causes a small geometrical transform on the image pattern, that is, we accept the relation $f(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{\theta}) = f(\boldsymbol{x}, \boldsymbol{\theta} + \boldsymbol{\psi})$. Therefore, small perturbation of an image caused by parameter-perturbation is replaced to geometrical perturbation of the image, that is,

$$f(\boldsymbol{x}, \boldsymbol{\theta} + \boldsymbol{\psi}) = f(\boldsymbol{x} + \boldsymbol{\delta}) = f(\boldsymbol{x}) + \boldsymbol{\delta}^\top \nabla f(x, y), \tag{6.3}$$

where $\boldsymbol{\delta}$ is a perturbation vector. Since

$$\int_{\mathbb{R}^2} f f_x d\boldsymbol{x} = 0, \int_{\mathbb{R}^2} f f_y d\boldsymbol{x} = 0, \int_{\mathbb{R}^2} f_x f_y d\boldsymbol{x} = 0, \tag{6.4}$$

for images $g(\boldsymbol{x}) = f(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{t})$ with a small perturbation affine transform $\boldsymbol{A}$ and a small translation vector $\boldsymbol{t}$, we can assume the relation

$$g(\boldsymbol{x}) = a_0 f(\boldsymbol{x}) + a_1 \partial_x f(\boldsymbol{x}) + a_2 \partial_y f(\boldsymbol{x}). \tag{6.5}$$

Equation (6.5) implies that the number of independent images among the collected images,

$$L(f) = \{f_{ij} | f_{ij}(\boldsymbol{x}) = \lambda f(\boldsymbol{A}_i \boldsymbol{x} + \boldsymbol{t}_j)\}_{i,j=1}^{p,q}, \tag{6.6}$$

is three. We can use the first three principal vectors of $\boldsymbol{L}_f$ as the local basis for image expression. We call this property the local linear property. Figure 6.1(b) shows the projection of the input image g to the three-dimensional local subspace.

6.4.2 Three-Dimensional Image

Setting the Hilbert space H to be the space of patterns, we assume that the inner product (f, g) is defined in H. Let $f \in H$ and P be a pattern and an operator for a class, respectively. We then define the class $\mathcal{C} = \{f \,|\, Pf = f,\, P^*P = I\}$. For recognition, we construct P for $f \in \mathcal{C}$ while minimising $E[\|f - Pf\|_2]$ with respect to $P^*P = I$, where $f \in \mathcal{C}$ is the pattern for a class, I is the identity operator and E is the expectation in H. This methodology is known as the subspace method [36, 65]. For the practical calculation of P, we adopt the Karhunen-Loeve expansion for the construction of the eigenspace.

We deal with images $f(\boldsymbol{x})$ defined in the three-dimensional Euclidean space $\boldsymbol{x} = (x, y, z)^\top \in \mathbb{R}^3$. We assume that a small perturbation of the parameter causes a small geometrical transform of the image pattern, that is, we assume the relation $f(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{\theta}) = f(\boldsymbol{x}, \boldsymbol{\theta} + \boldsymbol{\psi})$. Therefore, a small perturbation of an image caused by parameter perturbation is replaced with a geometrical perturbation of the image, that is,

$$f(\boldsymbol{x}, \boldsymbol{\theta} + \boldsymbol{\psi}) = f(\boldsymbol{x} + \boldsymbol{\delta}) = f(\boldsymbol{x}) + \boldsymbol{\delta}^\top \nabla f(x, y), \tag{6.7}$$

where $\boldsymbol{\delta} \in \mathbb{R}^3$ is a perturbation vector. Setting f_x, f_y and f_z to $\partial_x f(\boldsymbol{x})$, $\partial_y f(\boldsymbol{x})$ and $\partial_z f(\boldsymbol{x})$, respectively, since

$$\int_{\mathbb{R}^3} f f_x d\boldsymbol{x} = 0, \int_{\mathbb{R}^3} f f_y d\boldsymbol{x} = 0, \int_{\mathbb{R}^3} f f_z d\boldsymbol{x} = 0, \tag{6.8}$$

$$\int_{\mathbb{R}^3} f_x f_y d\boldsymbol{x} = 0, \int_{\mathbb{R}^3} f_y f_z d\boldsymbol{x} = 0, \int_{\mathbb{R}^3} f_z f_x d\boldsymbol{x} = 0, \tag{6.9}$$

for images $g(\boldsymbol{x}) = f(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{t})$ with a small perturbation affine transform $\boldsymbol{A}$ and a small translation vector $\boldsymbol{t}$, we can assume the relation

$$g(\boldsymbol{x}) = a_0 f + a_1 f_x + a_2 f_y + a_3 f_z, \ \boldsymbol{x} \in \mathbb{R}^3. \tag{6.10}$$

Equation (6.10) implies that the number of independent images among the collections of images,

$$L(f) = \{f_{ij} | f_{ij}(\boldsymbol{x}) = \lambda f(\boldsymbol{A}_i \boldsymbol{x} + \boldsymbol{t}_j)\}_{i,j=1}^{p,q} \tag{6.11}$$

is four, if the domain of the image is $\mathbb{R}^3$. We can use the first four principal vectors of $L(f)$ as the local basis for image expression for a three-dimensional image. We call this property the local linear property and the space spanned by $\{f, f_x, f_y, f_z\}$ the local eigenspace. Figure 6.1(b) shows the projection of the input image g to the three-dimensional local subspace.

6.5 Affine Transformation

6.5.1 Two-Dimensional Image

In an affine transformation, by translating the centre of gravity of an image to the centre point of the image, we can omit the translation in the estimation process of the affine transform. Therefore, we consider rotation, scaling and shearing. Furthermore, assuming a small affine transform with a small angle

ψ, small scaling factors λ_x and λ_y, and small shear ratios s_x and s_y, we have an affine transform given by the transform matrices such that

$$\boldsymbol{A}_1 = \boldsymbol{I} + \begin{pmatrix} 0 & -\psi \\ \psi & 0 \end{pmatrix} = \boldsymbol{I} + \boldsymbol{R}, \tag{6.12}$$

$$\boldsymbol{A}_2 = \boldsymbol{I} + \begin{pmatrix} \lambda_x & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \lambda_y \end{pmatrix} = \boldsymbol{I} + \boldsymbol{\Lambda}_x + \boldsymbol{\Lambda}_y, \tag{6.13}$$

$$\boldsymbol{A}_3 = \boldsymbol{I} + \begin{pmatrix} 0 & s_x \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ s_y & 0 \end{pmatrix} = \boldsymbol{I} + \boldsymbol{S}_x + \boldsymbol{S}_y, \tag{6.14}$$

respectively. Combining $\boldsymbol{A}_1, \boldsymbol{A}_2$ and $\boldsymbol{A}_3$, we can define all affine transforms except translation. For $\gamma_i = \{0, 1\},\ i = 1, 2, \ldots, 5$, multiplying these three matrices and ignoring higher-order terms than the first order, we have an affine transform matrix

$$\boldsymbol{A} = \boldsymbol{I} + \gamma_1 \boldsymbol{R} + + \gamma_2 \boldsymbol{\Lambda}_x + \gamma_3 \boldsymbol{\Lambda}_y + \gamma_4 \boldsymbol{S}_x + \gamma_5 \boldsymbol{S}_y = \boldsymbol{I} + \bar{\boldsymbol{A}}, \tag{6.15}$$

where the rotation, scaling and shear transform are commutative since their transform matrices consist of small-value elements. Here, $\sum_{i=1}^{5} \gamma_i$ represents the sum of coefficients in the target affine transform.

6.5.2 Three-Dimensional Image

To avoid the estimation of a translation, we set the origin of the coordinates to be the centre of an image. We assume small rotations around the x, y and z axes of angles ϕ_1, ϕ_2 and ϕ_3, given by the transform matrices such that

$$\boldsymbol{R}_x = \boldsymbol{I} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -\phi_1 \\ 0 & \phi_1 & 0 \end{pmatrix} = \boldsymbol{I} + \boldsymbol{R}'_x, \tag{6.16}$$

$$\boldsymbol{R}_y = \boldsymbol{I} + \begin{pmatrix} 0 & 0 & \phi_2 \\ 0 & 0 & 0 \\ -\phi_2 & 0 & 0 \end{pmatrix} = \boldsymbol{I} + \boldsymbol{R}'_y, \tag{6.17}$$

$$\boldsymbol{R}_z = \boldsymbol{I} + \begin{pmatrix} 0 & -\phi_3 & 0 \\ \phi_3 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \boldsymbol{I} + \boldsymbol{R}'_z. \tag{6.18}$$

Multiplying $\boldsymbol{R}_x, \boldsymbol{R}_y$ and $\boldsymbol{R}_z$, and ignoring terms of order larger than one, we have an arbitrary rotation expressed as

$$\boldsymbol{R}(\boldsymbol{\phi}) = \begin{pmatrix} 1 & -\phi_3 & \phi_2 \\ \phi_3 & 1 & -\phi_1 \\ -\phi_2 & \phi_1 & 1 \end{pmatrix} = \boldsymbol{I} + \boldsymbol{R}'_x + \boldsymbol{R}'_y + \boldsymbol{R}'_z = \boldsymbol{I} + [\boldsymbol{R}]_\times, \tag{6.19}$$

where $[\boldsymbol{R}]_\times$ is the outer-product operator of vector $\boldsymbol{\phi} = (\phi_1, \phi_2, \phi_3)^\top$.

For small scaling factors ϕ_4, ϕ_5 and ϕ_6, and small shearing ratios ϕ_7, ϕ_8, ϕ_9, ϕ_{10}, ϕ_{11} and ϕ_{12}, we have the scaling matrix and shearing matrix

$$\begin{pmatrix} 1+\phi_4 & 0 & 0 \\ 0 & 1+\phi_5 & 0 \\ 0 & 0 & 1+\phi_6 \end{pmatrix} = \boldsymbol{I} + \boldsymbol{\Lambda}, \tag{6.20}$$

$$\begin{pmatrix} 1 & \phi_7 & \phi_8 \\ \phi_9 & 1 & \phi_{10} \\ \phi_{11} & \phi_{12} & 1 \end{pmatrix} = \boldsymbol{I} + \boldsymbol{S}, \tag{6.21}$$

respectively.

Combining these rotation, scaling and shearing matrices in eqs. (11)-(6.21), we can define all affine transforms except translation. Multiplying these three matrices and ignoring terms of order larger than one, we have the affine transform matrix

$$\boldsymbol{A} = \boldsymbol{I} + [\boldsymbol{R}]_\times + \boldsymbol{\Lambda} + \boldsymbol{S} = \boldsymbol{I} + \boldsymbol{A}_\delta, \tag{6.22}$$

where the rotation, scaling and shear transforms are commutative since their transform matrices consist of only small-value elements. Here, $\boldsymbol{A}_\delta$ represents a small affine transform.

6.6 Neighbours of Template Image

For the reference image f and template image g in Hilbert space H, applying affine transforms $\{\boldsymbol{A}_i\}_{i=1}^N$ except for translation to f, we have the finite collection $\{f_i|\boldsymbol{A}_i\boldsymbol{x}\}_{i=1}^N$. For $0 < k \ll N$, let $\pi(i)$ be one-to-one injection from $1 \leq i \leq N$ to $1 \leq \pi(i) \leq k$ such that $\pi(i) \neq \pi(j)$ for $i \neq j$. Using $\pi(i)$, we define the k-neighbourhood $KN(g) \in L(f)$ of g. For a finite collection of images $\{f_i\}_{i=1}^N$, $KN(g)$ is a collection $\{f_{\pi(i)}\}_{i=1}^N$ that satisfies the inequalities

$$\|g - f_{\pi(1)}\|_2 \leq \|g - f_{\pi(2)}\|_2 \leq \cdots \leq \|g - f_{\pi(N)}\|_2 \tag{6.23}$$

where $\|\cdot\|_2$ is the L_2 metric on H.

6.7 Manifold Generation by Random Projection

We construct the image manifold of entries in the dictionary using the nearest-neighbour method. To reduce the time complexity of the nearest neighbourhood mesh on the image manifold, we adopt the random projection.

The random projection reduces the dimension of the discrete vector space while preserving both local and global topologies and geometries. The random projection satisfies the following theorem. For a set $X = \{\boldsymbol{x}_i\}_{i=1}^N$ of N points in d-dimensional Euclidean space, consider a mapping onto the set $\hat{X} = \{\hat{\boldsymbol{x}}_i\}_{i=1}^N$ in k-dimensional Euclidean space. For the vector $\boldsymbol{x} = (x_1, \ldots, x_d)^\top$, we define the Euclidean norm as $\|\boldsymbol{x}\|_2 = \left(\sum_{i=1}^d x_i\right)^{1/2}$. The Johnson-Lindenstrauss lemma indicates that there is a mapping approximately preserving the Euclid distance between two arbitrary points [83]. Setting $|\boldsymbol{x} - \boldsymbol{y}|_2$ to be the Euclidean distance between two points $\boldsymbol{x}$ and $\boldsymbol{y}$ in appropriate dimensional Euclidean space, the next Theorem is satisfied [55].

Theorem 6.1 *(Johnson-Lindenstrauss lemma). For a subspace with dimension* $\hat{d} \geq \hat{d}_0 = \frac{9 \log N}{\epsilon^2 - \frac{2}{3}\epsilon^3} + 1 = \mathcal{O}(\epsilon^{-2} \log N)$*, where* ϵ *is a real number such that* $0 < \epsilon < \frac{1}{2}$*, a set* X *of* N *d-dimensional points* $\{\boldsymbol{x}_i\}_{i=1}^N$ *and an integer* $\hat{d}$ *with* $\hat{d} \ll d$*, there exists a mapping* f *from* $\mathbb{R}^d$ *to* $\mathbb{R}^{\hat{d}}$ *such that*

$$(1 - \epsilon)|\boldsymbol{x}_j - \boldsymbol{x}_i|_2 \leq |\hat{\boldsymbol{x}}_j - \hat{\boldsymbol{x}}_i|_2 \leq (1 + \epsilon)|\boldsymbol{x}_j - \boldsymbol{x}_i|_2, \tag{6.24}$$

for all $i, j = 1, 2, \ldots, N$.

Therefore, setting $\boldsymbol{R}$ to be the random projection from $\mathbb{R}^d$ to $\mathbb{R}^{\hat{d}}$, Theorem 6.1 implies the relation

$$P(|\,|\boldsymbol{x} - \boldsymbol{y}|_2 - |\boldsymbol{R}\boldsymbol{x} - \boldsymbol{R}\boldsymbol{y}|_2\,| < \varepsilon) > 1 - \delta \tag{6.25}$$

for small positive constants ε and δ, where P is a probability distribution. To use these topological and geometrical properties for fast computation in the nearest neighbour method, we use sampled images to construct the image manifold of data in the dictionary.

Let $\mathbb{Z}^d$ be the integer grid in $\mathbb{R}^d$, Setting $\mathbf{D}$ and Δ to be a finite subset of $\mathbb{Z}^d$ and a positive number that defines the resolution of sampling, respectively, the distance

$$D(f, g) = \sqrt{\int_{\mathbb{R}^d} |f(\boldsymbol{x}) - g(\boldsymbol{x})|^2 d\boldsymbol{x}} \tag{6.26}$$

is approximately computed as

$$D(f, g) = \sqrt{\sum_{\boldsymbol{z} \in \mathbf{D} \subset \mathbb{Z}^d} |f(\Delta\boldsymbol{x}) - g(\Delta\boldsymbol{z}))|^2 \Delta}, \tag{6.27}$$

for functions $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ defined on $\mathbb{R}^d$.

By expressing $\{f(\Delta \boldsymbol{z})\}_{\boldsymbol{z}\in\mathbf{D}}$ and $\{g(\Delta \boldsymbol{z})\}_{\boldsymbol{z}\in\mathbf{D}}$ as finite vectors $\boldsymbol{f}$ and $\boldsymbol{g}$, respectively, eq. (6.27) is expressed as

$$D(f,g) = |\boldsymbol{f} - \boldsymbol{g}|_2 \tag{6.28}$$

if we set $\Delta = 1$. Using the random projection, the distance between $\boldsymbol{f}$ and $\boldsymbol{g}$ is computed as

$$D(f,g) \approx |\boldsymbol{R}\boldsymbol{f} - \boldsymbol{R}\boldsymbol{g}|, \tag{6.29}$$

for functions $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ defined on $\mathbb{R}^n$ such that

$$\int_{\mathbb{R}^n} |f(\boldsymbol{x})|^2 d\boldsymbol{x} < \infty, \quad \int_{\mathbb{R}^n} |g(\boldsymbol{x})|^2 d\boldsymbol{x} < \infty. \tag{6.30}$$

Therefore, by searching $KN(g)$ in $\hat{d}$-dimensional Euclidean space with the random projection, we obtain the discrete version of $KN(g)$. For practical computation, we adopt an efficient random projection [143].

6.8 Local Linear Method

6.8.1 Two-Dimensional Image

For a two-dimensional image, we introduce a method of reducing the number N of images in the dictionary. Using the local linear property of images in the image space, we first generate an image in a sparse dictionary [80, 119]. For the registration of a template g, using the generated image, we next estimate the small affine transform between the generated image and nearest neighbour of g in the dictionary. From the generated image and the estimated transform, the local linear method can generate new entries in the dictionary.Figure 6.1 shows a flow of this local linear method.

For image generation, we use the k nearest neighbours of g in the dictionary. Let $\{f^r\}_{i=1}^k \in \mathcal{L}$, be the rth neighbour of g. The random projection preserves the pairwise distances between vectorised images. Therefore, f^r is searched for in a random projected space. For a template $g(\boldsymbol{x})$, we assume $g(\boldsymbol{x}) = f^1(\boldsymbol{A}\boldsymbol{x}, \theta) + \epsilon$, where $\boldsymbol{A}$ gives best matching between g and f^1, and ϵ is a small difference between the reference pattern and the registered template pattern. Using the local linear property, we can approximate the space spanned by $\{u_i\}_{i=1}^3$ using one spanned by $\{g\} \cup \{f^r\}_{r=1}^3$ if the data space $\mathcal{L}$ is not extremely sparse. Using Gram-Schmidt orthonormalisation for f^1, f^2 and f^3, we obtain the basis $\{u_i\}_{i=1}^3$. Projecting the template to the space spanned by $\{u_i\}_{i=1}^3$, we obtain a new image,

$$g^* = \sum_{i=1}^{3} \alpha_i u_i, \tag{6.31}$$

from a triplet of pre-prepared entries in the dictionary. Here, α_i represent the coefficients of the linear combination.

For coefficients $\gamma_i \in \{0, 1\}$, the projected template image and its nearest neighbour $f^1(\boldsymbol{x}, \boldsymbol{\theta})$, using the Taylor expansion, we have

$$\begin{aligned} g^* = f^1(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{\theta}) = f^1(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{\theta}) = f^1((\boldsymbol{I} + \bar{\boldsymbol{A}})\boldsymbol{x}, \boldsymbol{\theta}) \\ = f^1(\boldsymbol{x}, \boldsymbol{\theta}) + (\bar{\boldsymbol{A}}\boldsymbol{x})^\top \nabla f^1(\boldsymbol{x}, \boldsymbol{\theta}) \end{aligned} \tag{6.32}$$

as an approximation. For transform matrix $\bar{\boldsymbol{A}}$, we have

$$(\bar{\boldsymbol{A}}\boldsymbol{x})^\top \nabla f^1(\boldsymbol{x}, \boldsymbol{\theta}) = g^* - f^1(\boldsymbol{x}, \boldsymbol{\theta}). \tag{6.33}$$

Setting

$$A(\boldsymbol{x}) = \left(x\frac{\partial f^1}{\partial y} - y\frac{\partial f^1}{\partial x}\right), B(\boldsymbol{x}) = \left(x\frac{\partial f^1}{\partial x}\right),$$
$$C(\boldsymbol{x}) = \left(y\frac{\partial f^1}{\partial y}\right), D(\boldsymbol{x}) = \left(y\frac{\partial f^1}{\partial x}\right), E(\boldsymbol{x}) = \left(x\frac{\partial f^1}{\partial y}\right),$$

we can represent the left side of eq. (6.33) as

$$\gamma_1\psi A(\boldsymbol{x}) + \gamma_2\lambda_x B(\boldsymbol{x}) + \gamma_3\lambda_y C(\boldsymbol{x}) + \gamma_4 s_x D(\boldsymbol{x}) + \gamma_5 s_y E(\boldsymbol{x}). \tag{6.34}$$

By solving eq. (6.33), we obtain the parameters.

However, the sum of coefficients $\sum_{i=1}^5 \gamma_i$ is greater than or equal to one although we have only one template. We adopt the line integral of eq. (6.34) for one template. Selecting different paths of a line integral, we obtain more than one independent equation. For the centre $\boldsymbol{\mu}$ of an image with radius $\{r_i\}_{i=1}^n$, we set

$$\mathcal{C}_i = \{\boldsymbol{x} | (\boldsymbol{x} - \boldsymbol{u})^\top \boldsymbol{I}(\boldsymbol{x} - \boldsymbol{u}) = r_i^2\},\ r_i \neq r_j \tag{6.35}$$

to be a path for the line integral. For the rotation matrices $\{\boldsymbol{R}_i\}_{i=1}^n$ with angle $\theta_i = \frac{\pi(i-1)}{2(n+1)}$ and $\boldsymbol{e}_x = (1, 0)^\top, \boldsymbol{e}_y = (0, 1)^\top$, we set

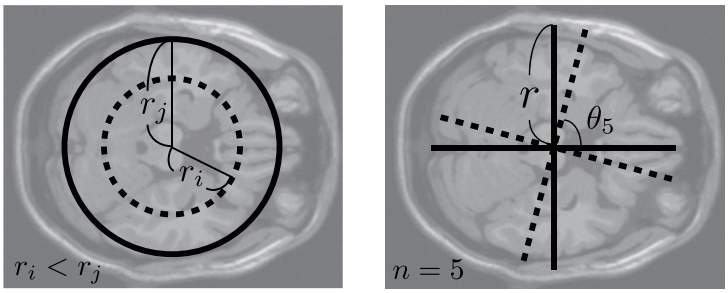
$$\mathcal{R}_i = \{\boldsymbol{\mu} + l\boldsymbol{R}_i\boldsymbol{e}_x, \boldsymbol{\mu} + l\boldsymbol{R}_i\boldsymbol{e}_y |\ -r \leq l \leq r\} \tag{6.36}$$

as another path for the line integral. Figures 6.3(a) and 6.3(b) show these two paths for the line integral. For $\mathcal{P}_i \in \{\mathcal{C}_i, \mathcal{R}_i\}_{i=1}^n$, we set coefficient vector $\boldsymbol{\chi}_i$ as

$$(\oint_{\mathcal{P}_i} A(\boldsymbol{x})d\boldsymbol{x}, \oint_{\mathcal{P}_i} B(\boldsymbol{x})d\boldsymbol{x}, \oint_{\mathcal{P}_i} C(\boldsymbol{x})d\boldsymbol{x}, \oint_{\mathcal{P}_i} D(\boldsymbol{x})d\boldsymbol{x}, \oint_{\mathcal{P}_i} E(\boldsymbol{x})d\boldsymbol{x},). \tag{6.37}$$

Setting $\boldsymbol{\zeta} = (\gamma_1\psi, \gamma_2\lambda_x, \gamma_3\lambda_y, \gamma_4 s_x, \gamma_5 s_y)^\top$ and $h_i = \oint_{C_i}(g^* - f^1)d\boldsymbol{x}$, we have the relations $\chi_i \neq \chi_j$ and $h_i \neq h_j$ for $i \neq j$. Therefore, we can estimate the parameters as a solution of

$$(\boldsymbol{\chi}_1^\top, \boldsymbol{\chi}_2^\top, \ldots, \boldsymbol{\chi}_5^\top)^\top \boldsymbol{\zeta} = (\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_5)^\top. \tag{6.38}$$



(a) (b)

Figure 6.3: Example of two paths for line integral. (a) and (b) show circular paths and line paths, respectively. In (a), the solid line and dashed lines represent circles $\mathcal{C}_i$ and $\mathcal{C}_j$, respectively. Here, we set $r_i < r_j$. In (b), the solid and dashed lines represent $\mathcal{R}_0$ and $\mathcal{R}_5$, respectively.

6.8.2 Three-Dimensional Image

Using the local linear property of images in the image space, we first generate an image in a sparse dictionary. To register a template g, using the generated image g^*, we next estimate the small affine transform between the generated image g^* and the nearest neighbour f^1 of g in the dictionary. From the generated image and the estimated transform, the local linear method can generate new entries in the dictionary. Figure 6.1 shows a flow of this local linear method.

For image generation, we use the k nearest neighbours of g in the dictionary. Let $\{f^i\}_{i=1}^k \in \mathcal{L}(g)$, be the ith neighbour of g. The random projection preserves the pairwise distances between vectorised images. Therefore, f^i is searched for in a random projected space. For a template $g(\boldsymbol{x})$, we assume $g(\boldsymbol{x}) = f^1(\boldsymbol{A}\boldsymbol{x}, \theta) + \epsilon$, where $\boldsymbol{A}$ gives the best matching between g and f^1, and ϵ is a small difference between the reference pattern and the registered template pattern. For three-dimensional images, using the local linear property, we can approximate the space spanned by $\{u_i\}_{i=1}^4$ using the space spanned by $\{g\} \cup \{f^i\}_{i=1}^4$ if the data space $\mathcal{L}(g)$ is not extremely sparse. Using Gram-Schmidt orthonormalisation for $\{f^i\}_{i=1}^4$, we obtain the basis $\{u_i\}_{i=1}^4$. Projecting the template to the space spanned by $\{u_i\}_{i=1}^4$, we obtain a new image,

$$g^* = \sum_{i=1}^{4} b_i u_i, \tag{6.39}$$

from a triplet of preprepared entries in the dictionary. Here, $\{b_i\}_{i=1}^4$ represents the coefficients of the linear combination.

For the projected template image and its nearest neighbour $f^1(\boldsymbol{x}, \boldsymbol{\theta})$, using the Taylor expansion, we have the relation

$$\begin{aligned} g^* = f^1(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{\theta}) = f^1(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{\theta}) = f^1((\boldsymbol{I} + \boldsymbol{A}_\delta)\boldsymbol{x}, \boldsymbol{\theta}) \\ = f^1(\boldsymbol{x}, \boldsymbol{\theta}) + (\boldsymbol{A}_\delta \boldsymbol{x})^\top \nabla f^1(\boldsymbol{x}, \boldsymbol{\theta}) \end{aligned} \tag{6.40}$$

if the higher order terms with respect ***delta*** is sufficiently small. For the transform matrix $\boldsymbol{A}_\delta$, we have the relation

$$(\boldsymbol{A}_\delta \boldsymbol{x})^\top \nabla f^1(\boldsymbol{x}, \boldsymbol{\theta}) = g^* - f^1(\boldsymbol{x}, \boldsymbol{\theta}). \tag{6.41}$$

Representing the left side of eq. (6.41) in terms of the variables that generate each transform, we can decompose the small affine transform between the reference and template. Using matrices $[\boldsymbol{R}]_\times, \boldsymbol{\Lambda}$ and $\boldsymbol{S}$, and coefficients $\gamma_i \in \{0, 1\}$, $i = 1, 2, \ldots, 9$, we can represent the left side of eq. (6.41) as

$$\boldsymbol{x}^\top \left(\begin{pmatrix} 0 & \gamma_3 & \gamma_2 \\ \gamma_3 & 0 & \gamma_1 \\ \gamma_2 & \gamma_1 & 0 \end{pmatrix} \circ [\boldsymbol{R}]_\times^\top + \operatorname{diag}(\gamma_4, \gamma_5, \gamma_6)\boldsymbol{\Lambda} + \operatorname{diag}(\gamma_7, \gamma_8, \gamma_9)\boldsymbol{S}^\top \right) \nabla f^1, \tag{6.42}$$

where $\boldsymbol{A} \circ \boldsymbol{B}$ is the Hadamard product of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. Furthermore, setting

$$\alpha_1 = y f_z^1 - z f_y^1, \alpha_2 = z f_x^1 - x f_z^1, \alpha_3 = y f_x^1 - x f_y^1, \tag{6.43}$$

$$\alpha_4 = x f_x^1, \alpha_5 = y f_y^1, \alpha_6 = z f_z^1, \tag{6.44}$$

$$\alpha_7 = y f_y^1, \alpha_8 = z f_z^1, \alpha_9 = x f_x^1, \alpha_{10} = z f_y^1, \alpha_{11} = x f_z^1, \alpha_{12} = y f_z^1, \tag{6.45}$$

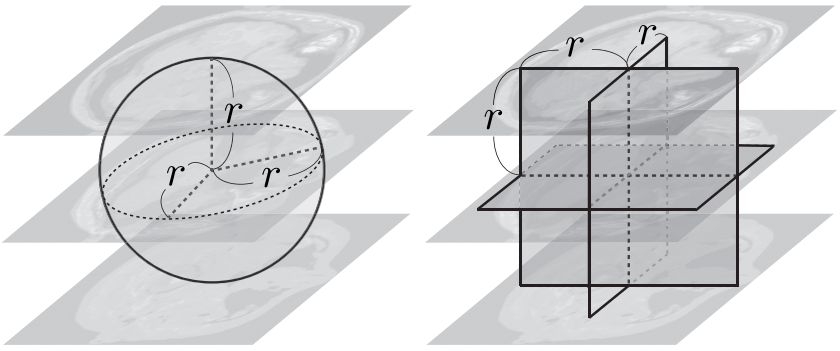
we rewrite eq. (6.41) as

$$\sum_{i=1}^{6} \gamma_i \alpha_i \phi_i + \sum_{j}^{3} (\alpha_{2(j-1)+7} \phi_{2(j-1)+7} + \alpha_{2(j-1)+8} \phi_{2(j-1)+8}) = g^* - f^1(\boldsymbol{x}, \boldsymbol{\theta}). \tag{6.46}$$

Equation (6.46) contains 12 unknowns in a single equation. The sum of coefficients $\sum_{i=1}^{6} \gamma_i + \sum_{i=7}^{9} 2\gamma_i$ is greater or equal to one even though we have only one template. We adopt the surface integration for eq. (6.46) for this template image. Selecting different surfaces of a surface integration, we obtain more than one independent equation. For the centre $\boldsymbol{\mu} = (\mu, \mu, \mu)^\top$ of a template image and radius $\{r_i\}_{i=1}^n$, $r_i \neq r_j$, we define surface of a sphere as

$$\mathcal{S}_3(r) = \{\boldsymbol{x} \mid \|\boldsymbol{x} - \boldsymbol{\mu}\|_2 = r\}. \tag{6.47}$$

For the centre $\boldsymbol{\mu} = (\mu, \mu, \mu)^\top$ of a template image, radius r, rotation angles $\boldsymbol{\phi}_i = (\phi_{i1}, \phi_{i2}, \phi_{i3})^\top$ and vectors $\boldsymbol{p}_1 = (x - \mu, y - \mu, -\mu)^\top$, $\boldsymbol{p}_2 = (-\mu, y -$



(a) (b)

Figure 6.4: Surfaces used for surface integration to obtain independent equations. (a) and (b) show surface $\mathcal{S}_3(r)$ of the sphere and surface $\mathcal{P}_3(r, 0)$ comprising three planes. Integration of the volume gives a equation for an image. The integration of different surfaces, such as a different spheres and orthogonal square planes, gives several independent equations for an image.

$\mu, z-\mu)^\top$ and $\boldsymbol{p}_3 = (x-\mu, -\mu, z-\mu)^\top$, we define the surface comprising three planes as

$$\mathcal{P}_3(r, \boldsymbol{\phi}) = \{\boldsymbol{\mu}+\boldsymbol{R}(\boldsymbol{\phi})\boldsymbol{p}_1, \boldsymbol{\mu}+\boldsymbol{R}(\boldsymbol{\phi})\boldsymbol{p}_2, \boldsymbol{\mu}+\boldsymbol{R}(\boldsymbol{\phi})\boldsymbol{p}_3 \mid \mu-r \leq x, y, z, \leq \mu+r\}. \tag{6.48}$$

For a set of radius $\{r_i\}_{i=1}^n$, we obtain sets of $\{\mathcal{S}_3(r_i)\}_{i=1}^n$ and $\{\mathcal{P}_3(r_i)\}_{i=1}^n$. We adopt $\{\mathcal{S}_3(r_i)\}_{i=1}^n$ and $\{\mathcal{P}_3(r_i)\}_{i=1}^n$ as surfaces $\{\Omega_i\}_{i=1}^n$ for the surface integration. Figure 6.4 shows the surfaces used for surface integration. For $\{\Omega_i\}_{i=1}^n$ and $\beta_{ij} = \int_{\Omega_i} \alpha_j(\boldsymbol{x})d\boldsymbol{x}, j = 1, 2, \ldots, 12$, we set the coefficient vector

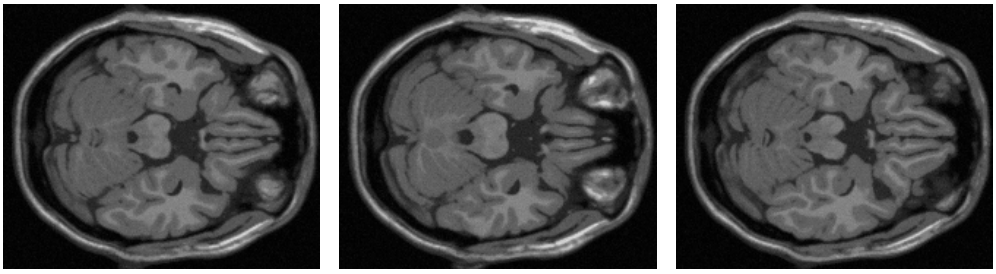
$$\boldsymbol{\chi}_i = (\beta_{i\,1}, \beta_{i\,2}, \ldots, \beta_{i\,12}). \tag{6.49}$$

Here, we have the relations $\chi_i \neq \chi_j$ and $h_i \neq h_j$ for $i \neq j$. Setting $n \geq \sum_{i=1}^6 \gamma_i + \sum_{i=7}^9 2\gamma_i$,

$$\begin{pmatrix} \beta_{1\,1} & \beta_{1\,2} & \ldots & \beta_{1\,12} \\ \beta_{2\,1} & \beta_{2\,2} & \ldots & \beta_{2\,12} \\ & & \vdots & \\ \beta_{n\,1} & \beta_{n\,2} & \ldots & \beta_{n\,12} \end{pmatrix} \boldsymbol{\zeta} = \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \end{pmatrix}. \tag{6.50}$$

where

$$\boldsymbol{\zeta} = (\gamma_1\phi_1, \gamma_2\phi_2, \ldots, \gamma_6\phi_6, \gamma_7\phi_7, \gamma_7\phi_8, \gamma_8\phi_9, \gamma_8\phi_{10}, \gamma_9\phi_{11}, \gamma_9\phi_{12})^\top, \tag{6.51}$$



(a) (b) (c)

(d) (e) (f)

Figure 6.5: Slice images and character images. (a)-(c) are slice images extracted from volume data obtained by MRI simulation of a human brain [37]. The size of the volume data is 181×217×181 pixels. The slice images (a), (b) and (c) are extracted from the $z = 50$, $z = 48$ and $z = 52$ planes, respectively. (d)-(f) are character images in a handwriting dataset [142]. The size of the character images is 127 × 128. In experiments, we embed (a)-(c) and (d)-(f) in 543×543 pixel and 272×272 pixel background images, respectively. The intensities of background images are 0.

and

$$h_i = \int_{\mathcal{S}_i} (g^* - f^1) d\boldsymbol{x}, \tag{6.52}$$

then, we can estimate the transforms as a solution to the linear system of equations.

6.9 Experiments

6.9.1 Two-Dimensional Image

To evaluate the accuracy of estimation of transform between two-dimensional images by our local linear method, we evaluate the estimation error of rota-

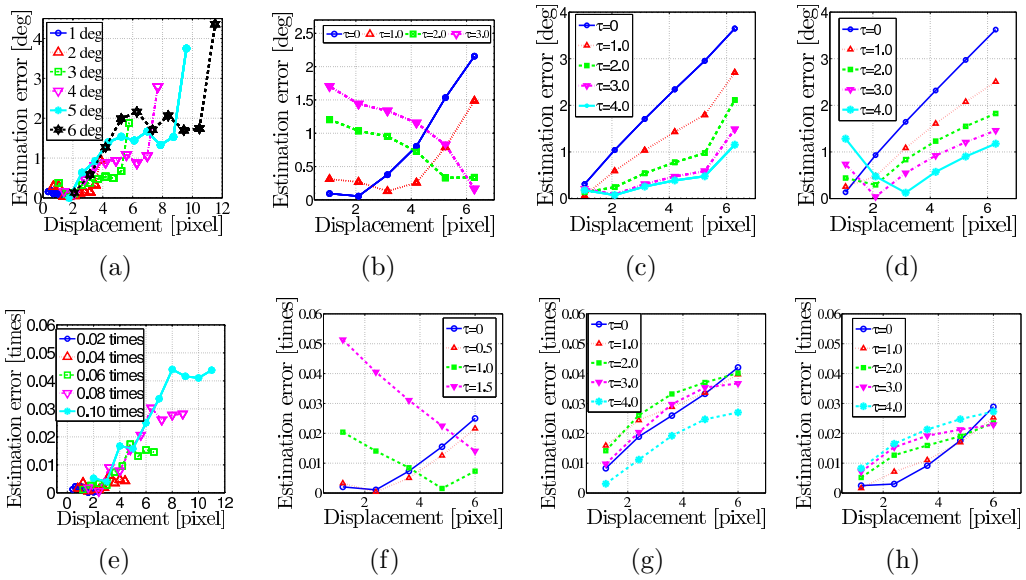


Figure 6.6: Accuracy of estimation of single transform for transformed slice images of human brain. Thee upper and lower rows represent the accuracy for rotation and scaling, respectively. The first column shows the accuracy for the transformed Fig. 6.5(a) with no filtering. The second, third and fourth columns show accuracy for the transformed Figs. 6.5(a), (b) and (c) with Gaussian filtering of the standard deviation τ.

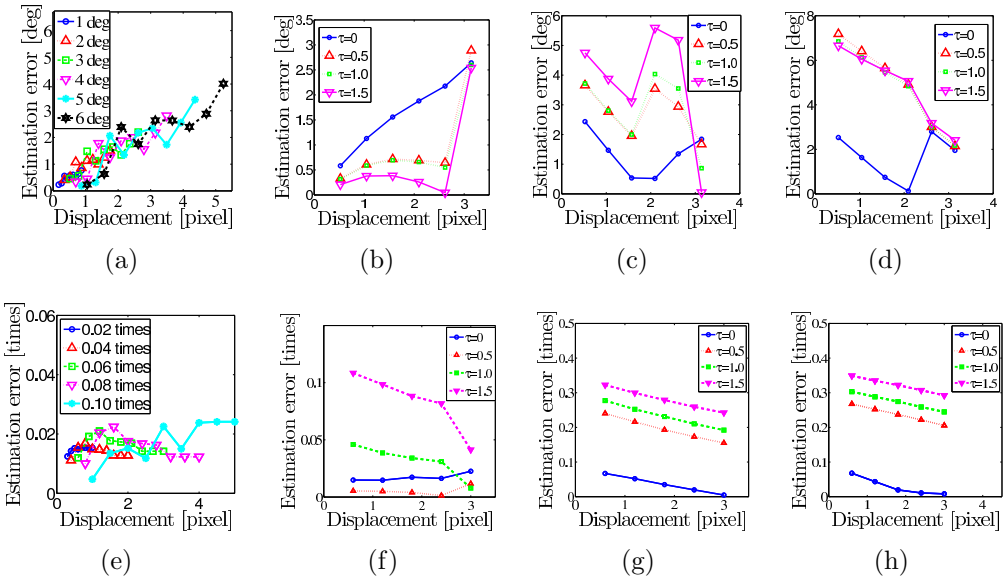


Figure 6.7: Accuracy of estimation of single transform for transformed character images. The upper and lower rows represent the accuracy for rotation and scaling, respectively. The first column shows the accuracy for the transformed Fig. 6.5(d) with no filtering. The second, third and fourth columns show the accuracy for the transformed Figs. 6.5(d), (e) and (f) with Gaussian filtering of standard deviation τ.

tion and isotropic scaling. We use two data sets: slice images extracted from volume data obtained by MRI simulation of a human brain [37] and images of a handwriting Chinese character [142]. Figures 6.5, (a)-(c) and (d)-(f) show the slice images and character images, respectively. The slice images are extracted from different planes of the volume data. The characters are written by different people. We select Figs 6.5. (a) and (d) as reference images for the slice images and Chinese characters, respectively.

First, we evaluate the estimation for single transform. For both two image sets, *i.e.*, slice images and character images, we use the following settings. For rotation and isotropic scaling, we generate two sets of transformed images with rotation angles of $-60, -48, \ldots, 60$ degrees and with scaling factor of $0.4, 0.6, \ldots, 1.4$ times, respectively. For rotation and isotropic scaling, the transform of the template are given by rotation angles of $1, 2, \ldots, 6$ degrees and scaling factor of $0.02, 0.04, \ldots, 0.10$ times, respectively. In the computation of the line integral, we adopt eqs. (6.35) and (6.36) as the paths for rotation and scaling, respectively. For the integration, we compute the summation of the absolute values of the left and right sides of eq. (6.35) for numerical stabilisation. We adopt radii $r = 20, 25, \ldots, 110$ for the path of the line integral. Furthermore, we apply Gaussian filtering to the generated images and template as preprocessing.

Second, we evaluate the estimation for multi transform. For the slice images, we generate rotated and isotropically scaled images with combinations of the rotation angles of $\theta \in \{-60, -48, \ldots, 60\}$ degrees and the scaling factors of $\lambda \in \{0.4, 0.45, \ldots, 1.4\}$ times. We set combinations of the rotation of $\theta \in \{-1, -2, \ldots, -6\}$ degrees and the scaling of $\lambda \in \{1.01, 1.02, 1.03\}$ times as transforms for the template. For the character images, we generate the rotated and isotropically scaled images with combinations of the rotation angles of $\theta \in \{-54, -48, \ldots, 54\}$ degrees and the scaling factors of $\lambda \in \{0.4, 0.5, \ldots, 1.4\}$ times. We set combinations of the rotation of $\theta \in \{-1, -2, \ldots, -4\}$ degrees and the scaling of $\lambda \in \{1.01, 1.02, 1 \ldots, 1.05\}$ times as transform for the template. For computational stability, we use five paths given by eq. (6.35) and five paths given by eq. (6.36) for estimation. We randomly select each of the five paths from radii $r = \{15, 20, 25, \ldots, 60\}$ for eq. (6.35) and five paths given by eq. (6.36).

Figures 6.6 and 6.7 summarise the first evaluation. Figure 6.8 summarises the second evaluation. Figures 6.6 (a) and (e), and 6.7(a) and (d) show that our method can accurately estimate a transform in the case of a small displacement with a sparse dictionary. Figures 6.6 (b) and (f), and 6.7(a) and (d) show that our method can accurately estimate a transform for a larger displacement by Gaussian filtering of a larger standard deviation. Figures 6.6(c), (d), (g) and (h) and 6.7(c), (d), (g) and (h) show that our method can

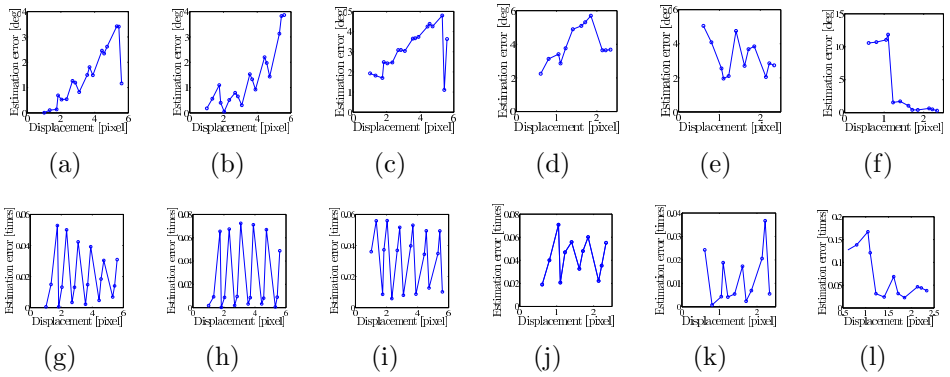


Figure 6.8: Accuracy of estimation of multi transform for slice images and character images. The upper and lower rows represent the accuracy for rotation and scaling, respectively. From left to right, each column represents the accuracy of parameter for the transformed Figs. 6.5(a), (b), (c), (d), (e) and (f). For the radius r of the integration path, rotation angle θ and scaling factor λ, we define the displacement as $\sqrt{(1+\lambda)^2 r^2 + \lambda^2 r^2}$.

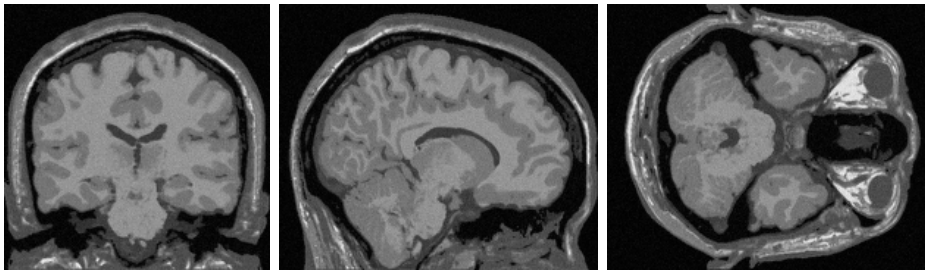
estimate a transform even for images which have small pattern perturbation. Figure 6.8 shows that our method can estimate more than one transform.

6.9.2 Three-Dimensional Image

Three experiments evaluate the performance of our local linear method for volumetric images. The first and second experiments show the accuracy of estimation for a single transform and multiple transforms, respectively. The third experiment shows the robust estimation of templates with small pattern perturbations.

The first and second experiments use volumetric data obtained by MRI simulation of human brain [37]. Figure 6.9 shows slice images of the volumetric data. Furthermore, for the first and second experiments, we generate smooth images from these slice images by linear filtering of the convolution with Gaussian kernel of standard deviation τ. For third experiment, we use volumetric spatiotemporal MRI lung data. [25]. Figure 6.9.2 shows a few frame of the volumetric spatiotemporal MRI lung data. In a sequence, the volumetric data gradually changes with the breathing of the patient. Table 6.1 summarises the parameters for the first and second experiments. Table 6.9.2 summarises the data for the third experiment.

Figure 6.11 shows the results of the first experiment for the estimation of rotation angle. In Figs. 6.11(a), (b) and (c), for displacements of less



(a) Coronal (b) Sagittal (c) Transverse

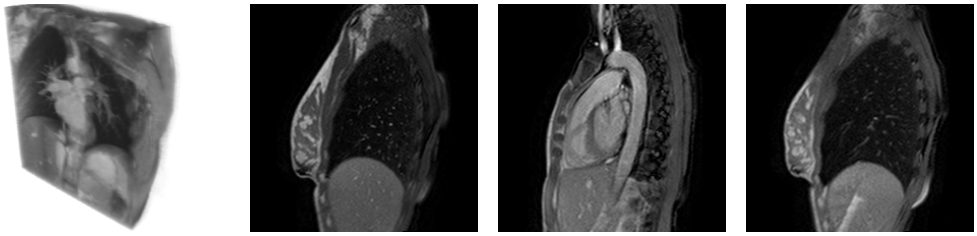
Figure 6.9: Slice images extracted from volumetric data. (a)-(c) Slice images extracted from a voxel image obtained by MRI simulation of a human brain [37]. The size of the voxel image is 181×217×181 voxels. The slice images (a), (b) and (c) are extracted from the $z = 45$, $x = 90$ and $y = 100$ planes, respectively. In experiments, we embed the voxel image in a background image of 308×308×308 voxels. The intensities of the background images are 0.

Table 6.1: Parameters for the first and second experiments using voxel image of human brain data.

$\boldsymbol{A}_\delta$	Pregeneration	Template	Filtering	Dimension	Lines
$\boldsymbol{R}_1$	$-60 < \phi_1 < 60$ step of 12	$1 < \phi_1 < 6$ step of 1	$0 < \tau < 5$ step of 0.1	$\hat{d} = 1024$	$\mathcal{S}_3(r)$ $10 < r < 110$ step of 10
$\boldsymbol{R}_2$	$-60 < \phi_2 < 60$ step of 12	$1 < \phi_2 < 6$ step of 1	$0 < \tau < 5$ step of 0.1	$\hat{d} = 1024$	$\mathcal{S}_3(r)$ $10 < r < 110$ step of 10
$\boldsymbol{R}_3$	$-60 < \phi_3 < 60$ step of 12	$1 < \phi_3 < 6$ step of 1	$0 < \tau < 5$ step of 0.1	$\hat{d} = 1024$	$\mathcal{S}_3(r)$ $10 < r < 110$ step of 10
$\boldsymbol{R}$	$-7 < \phi_1, \phi_2, \phi_3 < 7$ step of 7	$1 < \phi_1, \phi_2, \phi_3 < 3$ step of 1	$0 < \tau < 5$ step of 0.1	$\hat{d} = 1024$	$\mathcal{S}_3(r)$ $10 < r < 50$ step of 10

Table 6.2: Data for the the third experiment using the volumetric spatiotemporal data.

$\boldsymbol{A}_\delta$	Pregeneration with 22nd frame	Template with 22nd, 23rd, 24th and 34th frame	Filtering	Dimension	Lines
$\boldsymbol{R}_3$	$-60 < \phi_3 < 60$ step by 12	$1 < \phi_1 < 6$ step by 1	not used	$\hat{d} = 1024$	$\mathcal{S}_3(r)$ $10 < r < 20$ step by 5



(a) Image frame (b) Sagittal: $y = 9$ (c) Sagittal: $y = 29$ (d) Sagittal: $y = 41$

Figure 6.10: Volumetric spatiotemporal MRI lung data [25]. (a) Voxel image of a frame of a sequence. (b)-(d) Sagittal slices of the frame. The spatial and time resolutions of the data are 50×224×224 and 200, respectively. The time between frames is 331 ms. In the experiments, we embed a volumetric image of a frame on a background image of 316×316×316 voxels. Each voxel value in the background image is 0.

than 4 voxels, the estimation errors are smaller than 2.5 degrees if we use the surfaces $\{\mathcal{S}_3(r_i)\}_{i=1}^{10}$ for surface integration. Figures 6.11(d), (e) and (f) show that for displacements of less than 4 voxels, the estimation errors are smaller than 3.5 degrees if we use the surfaces $\{\mathcal{P}(r_i, \mathbf{0})\}_{i=1}^{10}$ for surface integration. In Figs. 6.11(g), (h), (i), (j), (k) and (l), for displacements of greater than 4 voxels in smooth images, our method estimates rotation angles with errors smaller than 1 degree.

The second experiment evaluates estimation errors for multiple transforms. Figure 6.12 shows the results of the second evaluation. In Fig. 6.12, the results show that the estimation of multiple transforms is unstable. Furthermore, the estimation errors are larger than 1 degree even for small displacements of one voxel. However, for smoothed images, the mean estimation error of the multiple transforms is about 1.5 degrees for the three rotation axes.

The third experiment evaluates the accuracy and robustness of estimation of rotation for a template with a small pattern perturbation. Figures 6.13(a) and (b) show the differences between the 22nd frame and the 23rd-200th frames of the four-dimensional data. Figure 6.13(c) shows the results of the estimation. In Fig. 6.13(c), curves represent absolute values of the estimation error plotted against the displacement caused by the rotation. For differences from $-\infty$ to -6.94 dB, the estimation errors are smaller than 1.5 degrees. Table 6.3 shows the difference between a template and generated g^*. In Table 6.3, the distance between the generated g^* and template is smaller than one between template and its nearest neighbour.

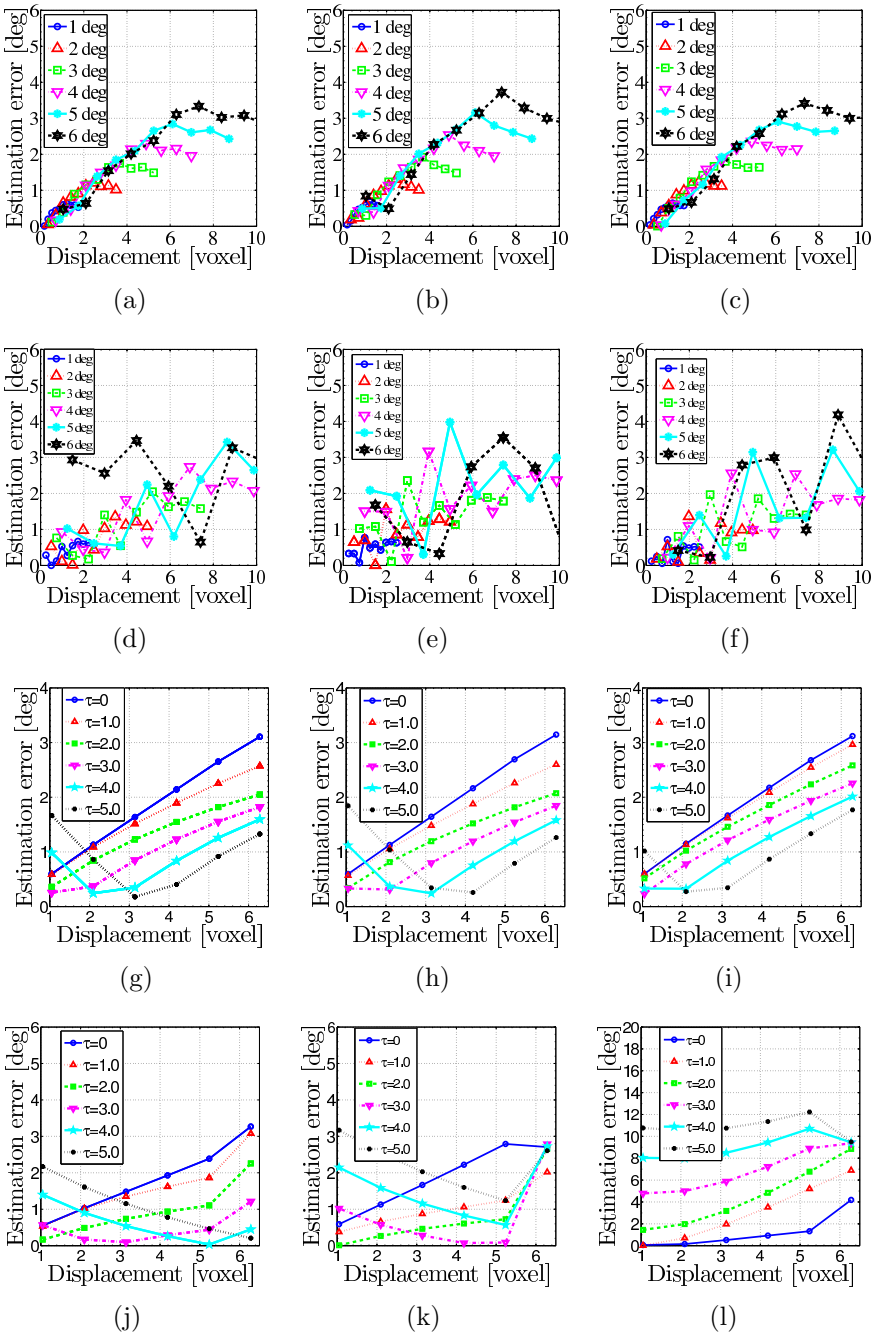
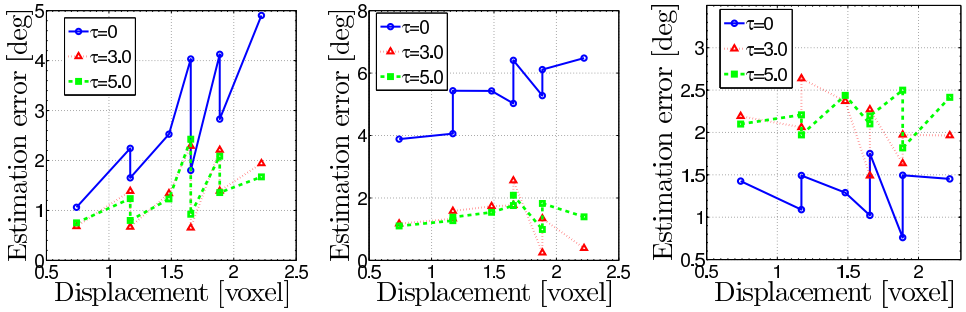


Figure 6.11: Accuracy of estimation for a spatial rotation. We estimate the rotation angles ϕ_1, ϕ_2 and ϕ_3 independently. The first, second and third columns represent the accuracy of estimation for rotation around the x, y and z axes, respectively. (a) and (d), (b) and (e), and (c) and (f) show the accuracy of estimation without Gaussian filtering. (g) and (j), (h) and (k), and (i) and (l) show the accuracy of estimation for smooth images, for the rotation around x, y and z axes, respectively. In the first and third rows and the second and fourth rows, we adopt $\mathcal{S}_3(r)$ and $\mathcal{P}_3(r, \boldsymbol{\phi})$ as the surfaces for the surface integration, respectively. Displacements are given by $r\phi_1, r\phi_2$ and $r\phi_3$.



(a) Rotation around x (b) Rotation around y (c) Rotation around z

Figure 6.12: Accuracy of estimation for multiple transforms. For the estimation, we adopt combinations of three rotations around the x, y and z axes. The left, middle and right graphs show the results of estimation for rotation around the x, y and z axes with Gaussian filtering with standard deviation τ, respectively. For the surface integration, we adopt surfaces $\{\mathcal{P}_i^1\}_{i=1}^n$. For the rotations around the x, y and z axes, the displacements are given by $r\sqrt{\phi_3^2+\phi_2^2}$, $r\sqrt{\phi_3^2+\phi_1^2}$ and $r\sqrt{\phi_1^2+\phi_2^2}$ with radius r in the surface integral, respectively.

Table 6.4 summarises the accuracy, the number of pregenerated images and the dimension of the search space in Figs. 6.11 and 6.12. The results in Fig. 6.11 imply that integration with the surfaces $\{\mathcal{S}_3(r_i)\}_{i=1}^{10}$ leads to more accurate and stable estimation than integration with the surfaces $\{\mathcal{P}(r_i, \mathbf{0})\}_{i=1}^{10}$ for the case of a rotation. For the estimation of a single transform, our method requires 16.7% of the number of pregenerated images of naive NNS. Furthermore, for the estimation of multiple transforms, our method requires 2.1% of the number of pregenerated images of the naive NNS. Moreover, our method reduces size of search space to 4.0×10^{-3}% . For both the estimations, the dimension of the search space is 3.5×10^{-3}% of the original dimension of the images. Moreover, the results of third experiment show that our method estimate transform for the template image with small pattern perturbation.

6.10 Summary

For two- and three-dimensional images, we first defined the local-linear property of the image manifold for a small geometrical perturbation. We then introduced an algorithm based on the local-linear property for two- and three-dimensional affine image registration to reduce the time and spatial

Table 6.3: Evaluation of approximation for generated new entries. We generate new entries for rotated images with small pattern perturbation. For a generation of a new entry, we use 4-neighbours of a template. As templates, we use rotated images of the 22nd, 23rd, 24th and 25th frame of data with angle ϕ_3. For a template g, we first compute the difference between g and its nearest neighbour in pregenerated images as $10\log_{10}(\|f^1 - g\|_2/\|g\|_2)$. Second, we compute the difference between g and a generated new entry g^* as $10\log_{10}(\|g^* - g\|_2/\|g\|_2)$. In this Table, the columns for the nearest neighbour (NN) and the local linear method (LLM) show the difference between f^1 and g and between g^* and g, respectively.

angle [degree]	Difference between f^1 and g[dB]									
	22nd		23rd		24th		25th		34th	
ϕ_3	NN	LLM	NN	LLM	NN	LLM	NN	LLM	NN	LLM
2	-4.77	-5.15	-4.85	-5.15	-4.78	-5.06	-4.39	-4.70	-4.60	-4.90
4	-3.19	-3.95	-3.31	-3.96	-3.26	-3.91	-3.21	-3.81	-3.25	-3.87
6	-2.57	-3.71	-2.70	-3.72	-2.66	-3.68	-2.69	-3.62	-2.68	-3.65

Table 6.4: Accuracy and compression ratio for volumetric data obtain by MRI simulation of human brain. First column shows given accuracy in the estimation. Second column shows necessary step sizes in pregeneration, which give the accuracy in first column, for the nearest neighbour search (NNS) and the local linear method. Third column shows dimensions of search space for NNS and LLM. Fourth column illustrates compression ration of the LLM compared with the NNS.

$\boldsymbol{A}_\delta$	Accuracy	Necessary step size		Dimension		Compression ratio
		NNS	LLM	Original	Search space	in pregeneration
$\boldsymbol{R}_1$	1 [degree]	2 [degree]	12 [degree]	29218112	1024	16.7 [%]
$\boldsymbol{R}_2$	1 [degree]	2 [degree]	12 [degree]	29218112	1024	16.7 [%]
$\boldsymbol{R}_3$	1 [degree]	2 [degree]	12 [degree]	29218112	1024	16.7 [%]
$\boldsymbol{R}$	1.5 [degree]	3 [degree]	12 [degree]			
	2.0 [degree]	4 [degree]	12 [degree]	29218112	1024	2.1 [%]
	1.5 [degree]	3 [degree]	12 [degree]			

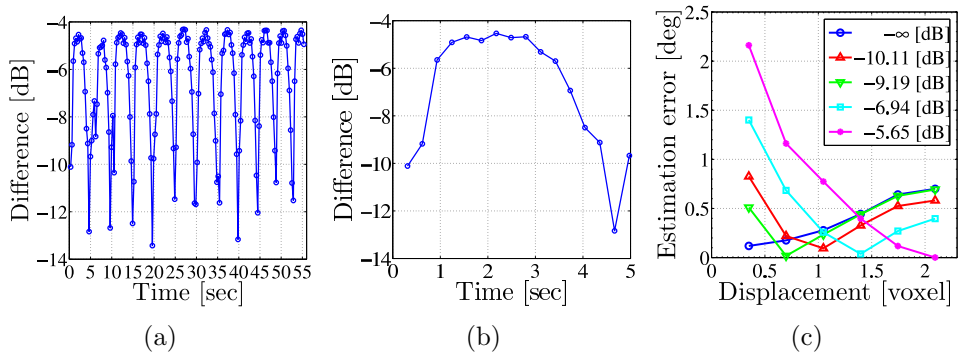


Figure 6.13: Estimation for a template with small pattern perturbation. (a) Difference between 22nd frame and 23rd-200th frames of four-dimensional MRI lung data. (b) Scaled-up graph of (a) showing difference between 22nd frame and 23rd-38th frames. (c) Accuracy of estimation for rotation angle ϕ_3 around the z axis. The differences between the 22nd frame and the 22nd, 23rd, 24th, 25th and 34th frames are $-\infty, -10.11, -9.19, -6.94$ and -5.65 [dB], respectively. For surface integration, we adopt the surface $\mathcal{S}_3(r)$. The displacement is given by $r\phi_3$.

complexity of computation. The algorithm first generates a new image for a template using a small number of images preproduced from the reference image. Second, using the new image, the proposed method finds a small affine transform between the new image and the best matching image in the dictionary. Finally, our method estimates transforms using the new image and its neighbours. This algorithm reduces the computational cost of preprocessing and the size of the images used in the nearest-neighbour search. In the numerical examples, we show that our method can accurately estimate single and multiple transforms using a small number of pregenerated images.

Chapter 7

Conclusions

In this dissertation, we introduced recognition methods for pattern in multilinear forms.

First, we introduced first-, second-, third- and Nth-order tensor representation for multidimensional data. For multiway data in these multilinear forms, using a multilinear projection, we introduced computational methods for signal processing such that principal component analysis and discrete cosine transform. For the approximation of principal component analysis for Nth-order tensors, we present N-dimensional discrete cosine transform as fast computational methods.

Second, we mathematically and experimentally show the effects of linear and bilinear dimension-reduction methods for pattern recognition methods of linear and bilinear forms. We defined four essential conditions for dimension reduction for image pattern recognition. By clarification of the nonexpansive mapping and topology-preserving mapping, we showed that only the topology-preserving mapping preserves distances and angles among data. The approximate preservation of distances and angles among data is the weak condition for dimension-reduction methods. Furthermore, we clarified that a classifier can achieve a higher recognition rate if we use nonexpansive mapping for preprocessing. This property is a common property to both linear and nonlinear methods. We concluded that the weak condition is only satisfied by the random projection among the linear dimension-reduction methods. For the bilinear recognition method, that is tensor subspace, tensor principal component analysis and two-dimensional discrete cosine transform give almost same effects in recognition rates. Furthermore, for linear recognition methods, two-dimensional discrete cosine transform gives almost the same effects of the random projection in recognition rates.

Third, we introduced tensor subspace method and mutual tensor subspace method for multiway data. By defining the tensor subspace of queries,

which belong to the same category, we can represent patterns of queries with geometrical perturbations. Tensor subspaces are found by tensor principal component analysis from sampled multiway data. We, then, define dissimilarity between two tensor subspaces. Using this dissimilarity, we developed the mutual tensor subspace method. In numerical experiments, we show validations of the subspace method and mutual tensor subspace method for three-way data. Using three-way data of gait patterns, we showed properties of computational methods for tensor principal component analysis, that is higher-order singular value decomposition and multilinear principal component analysis. The results showed that the computation by higher-order singular value decomposition gives almost the same results by multilinear principal component analysis. This property clarifies that iteration in the computation of principal component analysis is unnecessary. Furthermore, experimental results show that three-dimensional discrete cosine transform approximates third-order tensors in dimension reduction. Therefore, we conclude that N-dimensional discrete cosine transform is the fast approximation method for the computation of the tensor principal component analysis.

Fourth, we experimentally explore an essential feature in gradient fields in two-dimensional images, that is second-order tensors. From the results of our experiments, we have following observations. For the image pattern recognition, the discriminative feature is the edges of an image. In the context of the directional distribution, the dominant directional distributions represent the edges of a blurred image. For the image pattern recognition with the dominant directional distribution, the Wasserstein distance is an appropriate metric. As the acceptable approximation of the fast and accurate recognition with a pair of the dominant directional distributions and the Wasserstein distance, we can use the pair of the global directional distribution and L_1-norm. By the normalisation with the L_2-norm, the feature of histogram of oriented gradients possesses more discriminative distribution in a feature space than the features that represent probabilistic distributions of gradients.

Fifth, we introduced image registration methods, which are applications of the subspace method for two- and three-dimensional images. Our proposed algorithm, the local linear method, reduces the time and spatial complexity of computation of the dictionary-based image registration. The algorithm first generates a new image for a template using a small number of images preproduced from the reference image. In the numerical examples, we show that our method can accurately estimate single and multiple transforms using a small number of pregenerated images.

Bibliography

[1] Aase, S. O., Husoy, J. H. and Waldemar, P.: A critique of SVD-based image coding systems, Proc. *IEEE International Symposium on Circuits and Systems*, Vol. 4 (1999), 13–16.

[2] Achlioptas, D.: Database-friendly random projections: Johnson-Lindenstrauss with binary coins, *Journal of Computer and System Sciences*, Vol. 66 (2003), 671–687.

[3] Achlioptas, D. and McSherry, F.: Fast computation of low-rank matrix approximations, *Journal of the ACM*, Vol. 54 (2007).

[4] Agarwal, P. K., Har-Peled, S. and Yu, H.: Embeddings of surfaces, curves, and moving points in Euclidean space, Proc. *Annual Symposium on Computational Geometry*, (2007), 381–389.

[5] Ahmed, N., Natarajan, T. and Rao, K. R.: Discrete cosine transform, *IEEE Transactions on Computers*, Vol. C-23 (1974), 90–93.

[6] Ailon, N. and Liberty, E.: Almost optimal unrestricted fast Johnson-Lindenstrauss transform, *ACM Transactions on Algorithms*, Vol. 9 (2013), 21:1–21:12.

[7] Alexe, B., Petrescu, V. and Ferrari, V.: Exploiting spatial overlap to efficiently compute appearance distances between image windows, *Proc. Annual Conference on Neural Information Processing Systems*, (2011), 2735–2743.

[8] Allen, G.: Sparse higher-order principal components analysis, *Proc. International Conference on Artificial Intelligence and Statistics*, (2012), 27–36.

[9] Alpert, N. M., Bradshaw, , J. F., D., Kennedy and Correia, J. A.: The principal axes transformation–a method for image registration, *Journal of Nuclear Medicine*, Vol. 10 (1990), 1717–1722.

[10] Andreopoulos, A. and Tsotsos, J. K.: Efficient and Generalizable Statistical Models of Shape and Appearance for Analysis of Cardiac MRI, *Medical Image Analysis*, Vol. 12 (2008), 335–357.

[11] Anuta, P.: Spatial registration of multispectral and multitemporal digital imagery using fast Fourier transform techniques, *IEEE Transactions on Geoscience Electronics*, Vol. 8 (1970), 353–368.

[12] Artac, M., Jogan, M. and Leonardis, A.: Incremental PCA for online visual learning and recognition, *Proc. International Conference on Pattern Recognition, 2002.*, Vol. 3 (2002), 781–784.

[13] Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R. and Wu, A. Y.: An optimal algorithm for approximate nearest neighbor searching in Fixed Dimensions, Proc. *ACM-SIAM Symposium on Discrete Algorithms*, (1994), 573–582.

[14] Baraniuk, R. G. and Wakin, M. B.: Random projections of smooth manifolds, *Foundations of Computational Mathematics*, Vol. 9 (2009), 51–77.

[15] Barnea, D. I. and Silverman, H.: A class of algorithms for fast digital image registration, *IEEE Transactions on Computers*, Vol. C-21 (1972), 179–186.

[16] Benenson, R., Omran, M., Hosang, J. and Schiele, B.: Ten years of pedestrian detection, what have we learned?, Proc. *European Conference on Computer Vision Workshop on CVRSUAD*, (2014).

[17] Bian, X. and Krim, H.: Bi-sparsity pursuit for robust subspace recovery, *Proc. IEEE International Conference on Image Processing*, (2015), 3535–3539.

[18] Bigun, J.: *Vision with direction -a systematic introduction to image processing and computer vision*, Springer, 2006.

[19] Bingham, E. and Mannila, H.: Random projection in dimensionality reduction: Applications to image and text data, Proc. *International Conference on Knowledge Discovery and Data Mining*, (2001), 245–250.

[20] Björck, A. and Golub, G. H.: Numerical methods for computing angles between linear subspaces, *Mathematics of Computation*, Vol. 27 (1975), 579–594.

[21] Borg, I. and Groenen, P.: *Modern Multidimensional Scaling: Theory and Applications (2nd ed.)*, Springer, 2005.

[22] Borgefors, G., Ramella, G. and Baja, G. S. d.: Shape and topology preserving multi-valued image pyramids for multi-resolution skeletonization, *Pattern Recognition Letters*, Vol. 22 (2001), 741–751.

[23] Boser, E., Guyon, I. and Vapnik, V.: A training algorithm for optimal margin classifiers, Proc. *Workshop on Computational Learning Theory*, (1992), 144–152.

[24] Boutsidis, C., Garber, D., Karnin, Z. and Liberty, E.: Online principal components analysis, *Proc. Annual ACM-SIAM Symposium on Discrete Algorithms*, (2015), 887–901.

[25] Boye, D., Samei, G., Schmidt, J., Székely, G. and Tanner, C.: Population based modeling of respiratory lung motion and prediction from partial information, *In Proc. SPIE 8669, Medical Imaging 2013: Image Processing*, Vol. 8669 (2013).

[26] Boyer, K. L. and Ünsalan, C.: *Multispectral satellite image understanding: from land classification to building and road detection*, Springer, 2012.

[27] Bro-Nielsen, M. and Gramkow, C.: Fast fluid registration of medical images, in *The 4th International Conference on Visualization in Biomedical Computing*, Springer, 1996.

[28] Broit, C.: *Optimal registration of deformed images*, PhD thesis, University of Pennsylvania, 1981.

[29] Brown, L. G.: A survey of image registration techniques, *ACM Computing Surveys*, Vol. 24 (1992), 325–376.

[30] Burt, P. J. and Adelson, E. H.: The Laplacian pyramid as a compact image code, *IEEE Transactions on Communications*, Vol. 31 (1983), 532–540.

[31] Canny, J.: A computational approach to edge detection, *PAMI*, Vol. PAMI-8 (1986), 679–698.

[32] Capekm, M.: Optimisation strategies applied to global similarity based image registration methods, *Proc. the 7th International Congerence in Central Europoe on Computer Graphic*, (1999), 369–374.

[33] Christensen, G. E.: *Deformable shape models for anatomy*, PhD thesis, Washington University, 1994.

[34] Chum, O. and Matas, J.: Randomized RANSAC with $T_{d,d}$ test, *Image and Vision Computing*, Vol. 22 (2004), 837–842.

[35] Cichoki, A., Zdunek, R., Phan, A. H. and Amari, S.: *Nonnegative Matrix and Tensor Factorizations*, Wiley, 2009.

[36] Cock, K. D. and Moor, B. D.: Subspace angles between ARMA models, *Systems & Control Letters*, Vol. 46 (2002), 265–270.

[37] Cocosco, C., Kollokian, V., Kwan, R.-S. and Evans, A.: BrainWeb. Online Interface to a 3D MRI simulated brain database, *NeuroImage*, Vol. 5 (1997), 425.

[38] Cohen, N. and Shashua, A.: SimNets: a generalization of convolutional networks, *Proc. NIPS workshop on Deep Learning*, (2014).

[39] Cortes, C. and Vapnik, V.: Support-vector networks, *Machine Learning*, Vol. 20 (1995), 273–297.

[40] Csurka, G., Dance, C., Fan, L., Willamowski, J. and Bray, C.: Visual categorization with bags of keypoints, Proc. *ECCV Workshop on Statistical Learning in Computer Vision*, (2004), 1–22.

[41] Dalal, N. and Triggs, B.: Histograms of oriented gradients for human detection, Proc. *Computer Vision and Pattern Recognition*, (2005), 886–893.

[42] Dasgupta, S. and Gupta, A.: An elementary proof of the Johnson-Lindenstrauss lemma, Technical report, UC Berkeley, 1996.

[43] Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., Le, Q. V. and Ng, A. Y.: Large scale distributed deep networks, *Proc. NIPS*, (2012), 1232–1240.

[44] Ding, C. and Ye, J.: Two-dimensional singular value decomposition (2DSVD) for 2D maps and images, Proc. *SIAM International Conference on Data Mining*, (2005), 32–43.

[45] Dollár, P., Appel, R., Belongie, S. and Perona, P.: Fast feature pyramids for object detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 36 (2014), 1532–1545.

[46] Dollár, P., Tu, Z., Perona, P. and Belongie, S.: Integral channel features, Proc. *British Machine Vision Conference*, (2009).

[47] Duda, R. and Hart, P.: *Pattern Classification and Scene Analysis*, John Wiley and Sons, 1973.

[48] El-Zehiry, N. Y. and Grady, L.: Combinatorial optimization of the discretized multiphase Mumford-Shah functional, *International Journal of Computer Vision*, Vol. 104 (2013), 270–285.

[49] Ely, G., Aeron, S., Hao, N. and Kilmer, M. E.: 5D seismic data completion and denoising using a novel class of tensor decompositions, *Geophysics*, Vol. 80 (2015), V83–V95.

[50] Enomoto, H., Yonezaki, N. and Watanabe, Y.: Application of structure lines to surface construction and 3-dimensional analysis, in *Picture Engineering*, Vol. 6, Springer, 1982, 106–137.

[51] Everingham, M., Eslami, S. M. A., Gool, L. V., Williams, C. K. I., Winn, J. and Zisserman, A.: The Pascal Visual Object Classes Challenge: A Retrospective, *International Journal of Computer Vision*, Vol. 111 (2015), 98–136.

[52] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. and Zisserman, A.: The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results, http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index. html.

[53] Fei-Fei, L., Fergus, R. and Perona, P.: Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, *Computer Vision and Image Understanding*, Vol. 106 (2007), 59–70.

[54] Fisher, R. A.: The use of multiple measurements in taxonomic problems, *Annals of Eugenics*, Vol. 7 (1936), 179–188.

[55] Frankl, P. and Maehara, H.: The Johnson-Lindenstrauss lemma and the sphericity of some graphs, *Combinatorial Theory, Series B*, Vol. 44 (1988), 355–362.

[56] Fu, K. and Yu, T.: *Statistical Pattern Classification Using Contextual Information*, John Wiley and Sons, 1980.

[57] Fukui, K. and Maki, A.: Difference subspace and its generalization for subspace-based methods, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PP (2015), 1–1, In print.

[58] Fukui, K., Stenger, B. and Yamaguchi, O.: A framework for 3D object recognition using the kernel constrained mutual subspace method, *Proc. ACCV*, Vol. 3852 (2006), 315–324.

[59] Fukui, K., Yamaguchi, O., Suzuki, K. and Maeda, K.: Face recognition under variable lighting condition with constrained mutual subspace method, *Trans. IEICE (D-II)*, Vol. J82-D-II (1999), 613–620, (In Japanese).

[60] Georghiades, A. S., Belhumeur, P. N. and Kriegman, D. J.: From few to many: illumination cone models for face recognition under variable lighting and pose, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23 (2001), 643–660.

[61] Goh, A. and Vidal, R.: Clustering and dimensionality reduction on Riemannian manifolds, Proc. *IEEE Conference on Computer Vision and Pattern Recognition*, (2008), 1–7.

[62] Golub, G. H. and Van Loan, C. F.: *Matrix Computations*, The Johns Hopkins University Press, 1996.

[63] Grenander, U.: *Abstract Inference*, John Wiley, 1981.

[64] Hafner, D., Demetz, O. and Weickert, J.: Why is the census transform good for robust optic flow computation?, Proc. *Scale Space and Variational Methods in Computer Vision*, (2013).

[65] Hamm, J. and Lee, D. D.: Grassmann discriminant analysis: a unifying view on subspace-based learning, Proc. *International Conference on Machine Learning*, (2008), 376–383.

[66] Hao, Z., He, B., L. Chen and Yang, X.: A linear support higher-order tensor machine for classification, *IEEE Transactions on Image Processing*, Vol. 22 (2013), 2911–2920.

[67] Haralick, R.: Digital step edges from zero crossing of second directional derivatives, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 6 (1984), 58–68.

[68] Harandi, M. T., Salzmann, M. and Hartley, R.: From manifold to manifold: geometry-aware dimensionality reduction for PD matrices, Proc *European Conference on Computer Vision*, Vol. 8690 (2014), 17–32.

[69] Harris, C. and Pike, J.: 3D positional integration from image sequences, *Image and Vision Computing*, Vol. 6 (1988), 87–90.

[70] Hartley, R. I. and Zisserman, A.: *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2004.

[71] He, K., Zhang, Z., Ren, S. and Jian, S.: Spatial pyramid pooling in deep convolutional networks for visual recognition, Proc. *European Conference on Computer Vision*, (2014), 346–361.

[72] Helmke, U. and Moore, J. B.: Singular-value decomposition via gradient and self-equivalent flows, *Linear Algebra and Its Applications*, Vol. 169 (1992), 223–248.

[73] Hotelling, H.: Analysis of a complex of statistical variables into principal components, *Journal of Educational Psychology*, Vol. 24 (1933), 417–441.

[74] Hsu, C.-W. and Lin, C.-J.: A comparison of methods for multiclass support vector machines, *IEEE Transactions on Neural Networks*, Vol. 13 (2002), 415–425.

[75] Hua, G., Viola, P. A. and Drucker, S. M.: Face recognition using discriminatively trained orthogonal rank one tensor projections, Proc. *IEEE Conference on Computer Vision and Pattern Recognition*, (2007), 1–8.

[76] Iijima, T.: Theory of pattern recognition, *Electronics and Communications in Japan*, (1963), 123–134.

[77] Inoue, K., Hara, K. and Urahama, K.: Robust multilinear principal component analysis, *Proc. International Conference on Computer Vision*, (2009), 591–597.

[78] Itoh, H., Imiya, A. and Sakai, T.: Low-dimensional tensor principle component analysis, *Proc. International Conference on Computer Analysis of Images and Patterns, Part I*, Vol. 9256 (2015), 223–235.

[79] Itoh, H., Sakai, T., Kawamoto, K. and Imiya, A.: Dimension reduction methods for image pattern recognition, Proc. *International Workshop on Similarity-Based Pattern Recognition*, (2013), 26–42.

[80] Itoh, H., Sakai, T., Kawamoto, K. and Imiya, A.: Global image registration using random projection and local linear method, *In Proc. Internatinonal Conference on Computer Analysis of Images and Patterns*, (2013), 564–571.

[81] Itoh, H., Sakai, T., Kawamoto, K. and Imiya, A.: Topology-preserving dimension-reduction methods for image pattern recognition, Proc. *Scandinavian Conference on Image Analysis*, (2013), 195–204.

[82] Jégou, H., Perronnin, F., Douze, M., Sanchez, J., Perez, P. and Schmid, C.: Aggregating local image descriptors into compact codes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34 (2012), 1704–1716.

[83] Johnson, W. and Lindenstrauss, J.: Extensions of Lipschitz maps into a Hilbert space, *Contemporary Mathematics*, Vol. 26 (1984), 189–206.

[84] Karhunen, K.: Zur spektraltheorie stochastischer prozesse, *Annales Academiæ Scientiarum Fennnnicæ*, Vol. 34, 1946.

[85] Karlsson, A.: Nonexpanding maps, Busemann functions, and multiplicative ergodic theory, in *Rigidity in Dynamics and Geometry*, Springer, 2002, 283–294.

[86] Khan, F. S., Anwer, R. M., Weijer, van de J., Bagdanov, A. D., Vanrell, M. and Lopez, A. M.: Color attributes for object detection, in Proc. *Computer Vision and Pattern Recognition*, 2012.

[87] Kim, T.-K., Kittler, J. and Cipolla, R.: Discriminative learning and recognition of image set classes using canonical correlations, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29 (2007), 1005–1018.

[88] Kittler, J. and Young, P.: New approach to feature selection based on the Karhunen-Loeve expansion, *Pattern Recognition*, Vol. 5 (1973), 335–352.

[89] Klein, A., Andersson, J., Ardekani, B. A., Ashburner, J., Avants, B. B., Chiang, M.-C., Christensen, G. E., Collins, D. L., Gee, J. C., Hellier, P., Song, J. H., Jenkinson, M., Lepage, C., Rueckert, D., Thompson, P. M.,

Vercauteren, T., Woods, R. P., Mann, J. J. and Parsey, R. V.: Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration, *NeuroImage*, Vol. 46 (2009), 786–802.

[90] Kobayashi, T. and Otsu, Y.: Cone-restricted subspace methods, *Proc. International Conference on Pattern Recognition*, (2008).

[91] Kobayashi, T., Yoshikawa, F. and Otsu, N.: Cone-restricted kernel subspace methods, in *In International Conference on Image Processing*, 2010.

[92] Kohonen, T.: *Associative Memory: a System-Theoretical Approach*, Springer, 1977.

[93] Korman, S., Reichman, D., Tsur, G. and Avidan, S.: Fast-match: fast affine template matching, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, (2013), 1940–1947.

[94] Kotsia, I., Guo, W. and Patras, I.: Higher rank support tensor machines for visual recognition, *Pattern Recognition*, Vol. 45 (2012), 4192–4203.

[95] Kroonenberg, P. M. and Leeuw, J.: Principal component analysis of three-mode data by means of alternating least squares algorithms, *Psychometrika*, Vol. 45 (1980), 69–97.

[96] Kropatsch, W. G., Haxhimusa, Y., Pizlo, Z. and Langs, G.: Vision pyramids that do not grow too high, *Pattern Recognition Letters*, Vol. 26 (2005), 319–337.

[97] Kuglin, C. D. and Hines, D. C.: The phase correlation image alignment method, *Proc. Internatnional Conference of Cybernetics nas Society*, (1975), 163–165.

[98] Lampert, C. H., Blaschko, M. B. and Hofmann, T.: Efficient subwindow search: a branch and bound framework for object localization, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 31 (2009), 2129–2142.

[99] Lathauwer, L. D., Moor, B. D. and Vandewalle, J.: On the best rank-1 and rank-$(R_1, R_2, ..., R_N)$ approximation of higher-order tensors, *SIAM Journal on Matrix Analysis and Applications*, Vol. 21 (2000), 1324–1342.

[100] Lathauwer, L., Moor, B. and Vandewalle, J.: A multilinear singular value decomposition, *SIAM Journal on Matrix Analysis and Applications*, Vol. 21 (2000), 1253–1278.

[101] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P.: Gradient-based learning applied to document recognition, Proc. *IEEE*, Vol. 86 (1998), 2278–2324.

[102] Leese, J. A., Novak, C. S. and Clark, B. B.: An automated technique for obtaining cloud motion from geosynchronous Satellite data using cross correlation, *Journal of Applied Meteorology and Climatology*, Vol. 10 (1971), 118–132.

[103] Leibe, B. and Schiele, B.: Analyzing appearance and contour based methods for object categorization, Proc. *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2 (2003), 409–415.

[104] Liang, Z. and Shi, P.: An analytical algorithm for generalized low-rank approximations of matrices, *Pattern Recognition*, Vol. 38 (2005), 2213–2216.

[105] Lienhart, R. and Maydt, J.: An extended set of Haar-Like features for rapid object detection, Proc. *International Conference on Image Processing*, (2002), 900–903.

[106] Ling, H. and Okada, K.: An efficient earth mover's distance algorithm for robust histogram comparison, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 29 (2007), 840–853.

[107] Liu, H. and Ding, X.: Handwritten character recognition using gradient feature and quadratic classifier with multiple discrimination schemes, Proc. *International Conference on Document Analysis and Recognition*, (2005), 19–23.

[108] Loéve, M.: Functions aleatoires du second ordre, *Supplement to P. Lévy, Processus Stochastiques et Mouvement Brownien*, 1948.

[109] Longuet-Higgins, H. C.: A computer algorithm for reconstructing a scene from two projections, *Nature*, Vol. 293 (1981), 133–135.

[110] Lowe, D. G.: Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, Vol. 60 (2004), 91–110.

[111] Lu, H., Plataniotis, K. N. and Venetsanopoulos, A. N.: A survey of multilinear subspace learning for tensor data, *Pattern Recognition*, Vol. 44 (2011), 1540–1551.

[112] Lu, H., Plataniotis, K. and Venetsanopoulos, A.: MPCA: Multilinear principal component analysis of tensor objects, *IEEE Transactions on Neural Networks*, Vol. 19 (2008), 18–39.

[113] Lu, H., Plataniotis, K. and Venetsanopoulos, A.: Uncorrelated multilinear principal component analysis for unsupervised multilinear subspace learning, *IEEE Transactions on Neural Networks*, Vol. 20 (2009), 1820–1836.

[114] Maaten, L. J. P. V. d., Postma, E. O. and Herik, H. J. V. d.: Dimensionality reduction: A comparative review, Technical report, Tilburg University, 2009.

[115] Maeda, E. and Murase, H.: Multi-category classification by kernel based nonlinear subspace method, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 2 (1999), 1025–1028.

[116] Maeda, K.: From the subspace methods to the mutual subspace method, in *Computer Vision*, Vol. 285, Springer, 2010, 135–156.

[117] Maeda, K. and Watanabe, S.: A pattern matching method with local structure (in Japanese), Technical report, The Institute of Electronics, Information and Communication Engineers, 1984.

[118] Magen, A. : Dimensionality reductions that preserve volumes and distance to affine Spaces, and their algorithmic applications, Proc. *Randomization and Approximation Techniques in Computer Science*, Vol. 2483 (2002), 239–253.

[119] Mahajan, D., Huang, F.-C., Matusik, W., Ramamoorthi, R. and Belhumeur, P.: Moving gradients: a path-based method for plausible image interpolation, *ACM Transaction on Graphics*, Vol. 28 (2009), 1–11.

[120] Makihara, Y., Mannami, H., Tsuji, A., Hossain, M., Sugiura, K., Mori, A. and Yagi, Y.: The OU-ISIR gait database comprising the treadmill dataset, *IPSJ Trans. on Computer Vision and Applications*, Vol. 4 (2012), 53–62.

[121] Mardia, K. V. and Jupp, P. E.: *Directional Statistics*, John Wiley & Sons Ltd., 2000.

[122] Matousek, J.: On variants of the Johnson-Lindenstrauss lemma, *Random Structures & Algorithms*, Vol. 33 (2008), 142–156.

[123] Mika, S., Ratsch, G., Weston, J., Schölkopf, B. and Muller, K.: Fisher discriminant analysis with kernels, Proc. *IEEE Signal Processing Society Workshop. Neural Networks for Signal Processing IX*, (1999), 41–48.

[124] Mobahi, H., Collobert, R. and Weston, J.: Deep learning from temporal coherence in video, Proc. *International Conference on Machine Learning*, (2009).

[125] Moore, J. B., Mahony, R. E. and Helmke, U.: Numerical gradient algorithms for eigenvalue and singular value calculations, *SIAM Journal on Matrix Analysis and Applications*, Vol. 15 (1994), 881–902.

[126] Moradi, M., Abolmaesoumi, P. and Mousavi, P.: Deformable registration using scale space keypoints, *Proc. SPIE Medical Imagin*, Vol. 6144 (2006).

[127] Moshfeghi, M.: Elastic matching of multimodality medical images, *CVGIP: Graphical Models and Image Processing*, Vol. 53 (1991), 271–282.

[128] Mumford, D. and Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems, *Communications on Pure And Applied Mathematics*, Vol. 42 (1988), 577–685.

[129] Murase, H. and Nayar, S. K.: Illumination planning for object recognition using parametric eigenspace, *IEEE Transanctions on Pattern Analysis and Machine Intelligence*, Vol. 16 (1994), 1219–1227.

[130] Oja, E.: *Subspace Methods of Pattern Recognition*, Research Studies Press, 1983.

[131] Otsu, N.: *Mathematical Studies on Feature Extraction in Pattern Recognition*, PhD thesis, Electrotechnical Laboratory, 1981.

[132] Park, C. H. and Park, H.: Fingerprint classification using fast Fourier transform and nonlinear discriminant analysis, *Pattern Recognition*, Vol. 38 (2005), 495–503.

[133] Park, H. A. and Park, K. R.: Iris recognition based on score level fusion by using SVM, *Pattern Recognition Letters*, Vol. 28 (2007), 2019–2028.

[134] Pele, O. and Werman, M.: A linear time histogram metric for improved SIFT matching, Proc. *European Conference on Computer Vision*, (2008), 495–508.

[135] Pele, O. and Werman, M.: Fast and robust Earth Mover's Distances, Proc. *International Conference on Computer Vision*, (2009), 460–467.

[136] Pluim, J. P. W., Maintz, J. B. A. and Viergever, M. A.: Mutual-information-based registration of medical images: a survey, *Medical Imaging, IEEE Transactions on*, Vol. 22 (2003), 986–1004.

[137] Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P.: *Numerical Recipes in C*, Cambridge University Press, 1992.

[138] Pritchett, P. and Zisserman, A.: Wide baseline stereo matching, *Proc. ICCV*, (1998), 754–760.

[139] Rabin, J., Delon, J. and Gousseau, Y.: Circular Earth Mover's distance for the comparison of local features, Proc. *International Conference on Pattern Recognition*, (2008), 1–4.

[140] Roweis, S. T. and Saul, L. K.: Nonlinear dimensionality reduction by locally linear embedding, *Science*, Vol. 290 (2000), 2323–2326.

[141] Rubner, Y., Tomasi, C. and Guibas, L. J.: The Earth Mover's distance as a metric for image retrieval, *International Journal of Computer Vision*, Vol. 40 (2000), 99–121.

[142] Saito, T., Yamada, H. and Yamada, K.: On the data base ETL9 of handprinted characters in JIS Chinese characters and its analysis, *IEICE Transactions*, Vol. J68-D (1985), 757–764.

[143] Sakai, T. and Imiya, A.: Practical algorithms of spectral clustering: toward large-scale vision-based motion analysis, in *Machine Learning for Vision-Based Motion Analysis*, Springer, 2011, 3–26.

[144] Sakano, H. and Mukawa, N.: Kernel mutual subspace method for robust facial image recognition, *Proc. International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, Vol. 1 (2000), 245–248.

[145] Sakano, H., Mukawa, N. and Nakamura, T.: Kernel mutual subspace method and its application for object recognition, *Electronics and Communications in Japan (Part II: Electronics)*, Vol. 88 (2005), 45–53.

[146] Samaria, F. and Harter, A.: Parameterisation of a stochastic model for human face identification, Proc. *IEEE Workshop on Applications of Computer Vision*, (1994).

[147] Sargent, D., Chen, C.-I., Tsai, C.-M., Wang, Y.-F. and Koppel, D. K.: Feature detector and descriptor for medical images, *Proceedings of SPIE Medical Imaging*, Vol. 7259 (2009).

[148] Sarlos, T.: Improved approximation algorithms for large matrices via random projections, *Proc. EEE Symposium on Foundations of Computer Science*, (2006), 143–152.

[149] Schölkopf, B., Smola, A. and Müller, K.-R.: Nonlinear component analysis as a kernel eigenvalue problem, *Neural Computation*, Vol. 10 (1998), 1299–1319.

[150] Sergeev, S., Zhao, Y., George, M. and Okada, K.: Medical image registration using machine learning-based interest point detector, *Proceedings of SPIE Medical Imaging*, Vol. 8314 (2012).

[151] Shen, X., Krim, H. and Gu, Y.: Beyond union of subspaces: Subspace pursuit on Grassmann manifold for data representation, *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, (2016), 4079–4083.

[152] Sivic, J. and Zisserman, A.: Video Google: a text retrieval approach to object matching in videos, *Proc. IEEE International Conference on Computer Vision*, Vol. 2 (2003), 1470–1477.

[153] Sivic, J. and Zisserman, A.: Efficient visual search of videos cast as text retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31 (2009), 591–606.

[154] Studholme, C., Hill, D. L. and Hawkes, D. J.: Automated 3-D registration of MR and CT images of the head, *Medical Image Analysis*, Vol. 1, 163–175.

[155] Tao, D., Li, X., Wu, X., Hu, W. and Maybank, S. J.: Supervised tensor learning, *Knowledge and Information Systems*, Vol. 13 (2007), 1–42.

[156] Tao, D., Li, X., Wu, X. and Maybank, S.: Elapsed time in human gait recognition: a new approach, Proc. *International Conference on Acoustics, Speech and Signal Processing*, Vol. 2 (2006).

[157] Tenenbaum, J. B., Silva, V. d. and Langford, J. C.: A global geometric framework for nonlinear dimensionality reduction, *Science*, Vol. 290 (2000), 2319–2323.

[158] Torr, P. H. S. and Zisserman, A.: MLESAC: a new robust estimator with application to estimating image geometry, *Computer Vision and Image Understanding*, (2000), 138–156.

[159] Torr, P. H. S., Zisserman, A. and Maybank, S. J.: Robust detection of degenerate configurations while estimating the fundamental matrix, *Computer Vision and Image Understanding*, Vol. 71 (1998), 312–333.

[160] Tucker, L.: Some mathematical notes on three-mode factor analysis, *Psychometrika*, Vol. 31 (1966), 279–311.

[161] Turk, M. and Pentland, A.: Eigenfaces for recognition, *Journal of Cognitive Neuroscience*, Vol. 3 (1991), 71–86.

[162] Elsen, van den P., Pol, E.-J. and Viergever, M.: Medical image matching-a review with classification, *Engineering in Medicine and Biology Magazine, IEEE*, Vol. 12 (1993), 26–39.

[163] Vapnik, V. N.: *The Nature of Statistical Learning Theory*, Springer, 1995.

[164] Vapnik, V. and Lerner, A.: Pattern recognition using generalized portrait method, *Automation and Remote Control*, (1963), 774–778.

[165] Vedaldi, A., Gulshan, V., Varma, M. and Zisserman, A.: Multiple kernels for object detection, Proc. *Computer Vision and Pattern Recognition*, (2009), 606–613.

[166] Vempala, S. S.: *The Random Projection Method*, American Mathematical Society, 2004.

[167] Venna, J. and Kaski, S.: Local multidimensional scaling, *Neural Networks*, Vol. 19 (2006), 889–899.

[168] Vidal, R., Yi, M. and Sastry, S.: Generalized principal component analysis (GPCA), *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27 (2005), 1945–1959.

[169] Siebenthal, von M., Cattin, P., Gamper, U., Lomax, A. and Székely, G.: 4D MR imaging using internal respiratory gating, Proc. *MICCAI*, (2005), 336–343.

[170] Vondrick, C., Khosla, A., Malisiewicz, T. and Torralba, A.: HOGgles: visualizing object detection features, Proc. *International Conference on Computer Vision*, (2013).

[171] Wakabayashi, T., Tsuruoka, S., Kimura, F. and Miyake, Y.: Accuracy improvement through increased feature size in handwritten numeral recognition, *The transactions of the Institute of Electronics, Information and Communication Engineers (Japanese)*, Vol. 77 (1994), 2046–2053.

[172] Wang, H. and Ahuja, N.: Compact representation of multidimensional data using tensor randk-one decomposition, Proc. *International Conference on Pattern Recognition*, Vol. 1 (2004), 44–47.

[173] Wang, Y. and Gong, S.: Tensor discriminant analysis for view-based object recognition, Proc. *International Conference on Pattern Recognition*, (2006), 33–36.

[174] Wasserstein, L. N.: Markov processes over denumerable products of spaces describing large systems of automata, *Problems of Information Transmission*, Vol. 5 (1969), 47–52.

[175] Watanabe, S.: Karhunen-Loeve expansion and factor analysis, Proc. *Transactions of the Fourth Prague Conference on Information Theory, Statistical Decision Functions, Random Processes*, (1965), 635–660.

[176] Watanabe, S., Lambert, P. F., Kulikowski, C. A., Buxton, J. L. and Walker, R.: Evaluation and selection of variables in pattern recognition, *Computer and Information Science II*, (1967), 91–122.

[177] Watanabe, S. and Pakvasa, N.: Subspace method of pattern recognition, In *Proc. of the 1st International Joint Conference of Pattern Recognition*, (1973).

[178] Watanabe, T., Takimoto, E., Amano, K. and Maruoka, A.: Random projection and its application to learning, Proc. *Workshop on Randomness and Computation*, (2005), 3–4.

[179] Williams, C. K. I.: On a connection between kernel PCA and metric multidimensional scaling, *Machine Learning*, Vol. 46 (2002), 11–19.

[180] Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S. and Ma, Y.: Robust face recognition via sparse representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31 (2009), 210–227.

[181] Yamaguchi, O., Fukui, K. and Maeda, K.: Face recognition using temporal image sequence, *In Proc. Third IEEE International Conference on Automatic Face and Gesture Recognition*, (1998), 318–323.

[182] Yan, J., Zhang, X., Lei, Z., Liao, S. and Li, S. Z.: Robust multiresolution pedestrian detection in traffic scenes, Proc. *Computer Vision and Pattern Recognition*, (2013), 3033–3040.

[183] Yan, S., Xu, D., Yang, Q., Zhang, L., Tang, X. and Zhang, H. J.: Multilinear discriminant analysis for face recognition, *IEEE Transactions on Image Processing*, Vol. 16 (2007), 212–220.

[184] Yang, J., Zhang, D., Frangi, A. F. and Yang, J.-Y.: Two-dimensional PCA: a new approach to appearance-based face representation and recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26 (2004), 131–137.

[185] Ye, J., Janardan, R. and Qi, L.: GPCA: An efficient dimension reduction scheme for image compression and retrieval, Proc. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2004), 354–363.

[186] Yoshimura, S. and Kanade, T.: Fast template matching based on the normalized correlation by using multiresolution eigenimages, *Proc. IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, Vol. 3 (1994), 2086–2093.

Publications

Journal Papers (refereed)

1. Hayato Itoh, Atsushi Imiya, Tomoya Sakai: Pattern recognition in multilinear space and its applications: mathematics, computational algorithms and numerical validations. *Machin Vision and Applications*, vol. 27, pp.1259-1273, 2016.

2. Hayato Itoh, Atsushi Imiya, Tomoya Sakai: Dimension reduction and construction of feature space for image pattern recognition. *Journal of Mathematical Imaging and Vision*, vol. 56, pp. 1-31, 2016.

International Conferences (refereed)

1. Hayato Itoh, Atsushi Imiya, Tomoya Sakai: Approximation of n-way principal component analysis for organ data. *Proc. ACCV workshop on Mathematical and Computational Methods in Biomedical Imaging and Image Analysis*, in Print, 2016.

2. Hayato Itoh, Atsushi Imiya, Tomoya Sakai:Classification of volumetric data using multiway data analysis. *Proc. International Workshops on Structural and Syntactic Pattern Recognition and Statistical Techniques in Pattern Recognition*, Lecture Notes in Computer Science, vol.10029, pp. 231-240, 2016.

3. Hayato Itoh, Atsushi Imiya, Tomoya Sakai: Mathematical aspects of tensor subspace method. *Proc. International Workshops on Structural and Syntactic Pattern Recognition and Statistical Techniques in Pattern Recognition*, Lecture Notes in Computer Science, vol.10029, pp. 37-,48 , 2016.

4. Hayato Itoh, Atsushi Imiya, Tomoya Sakai: Volumetric image pattern recognition using three-way principal component analysis. *Proc. MIC-*

CAI workshop on Spectral and Shape Analysis in Medical Imaging , Lecture Notes in Computer Science, vol.10126, pp. 103-117, 2016.

5. Hayato Itoh, Atsushi Imiya, Tomoya Sakai: Discriminative properties in directional distributions for image pattern recognition. *Proc. Pacific Rim Symposium on Image and Video Technology*, Lecture Notes in Computer Science, vol. 9431, pp. 617-630, 2015.

6. Shum Inagaki, Hayato Itoh, Atsushi Imiya: Simultaneous frame-rate up-conversion of image and optical flow Sequences. *Proc. International Conference on Computer Vision Theory and Applications*, pp. 68-75, 2015.

7. Hayato Itoh, Atsushi Imiya, Tomoya Sakai: Low-dimensional tensor principle component analysis. *Proc. International Conference on Analysis of Images and Patterns*, Lecture Notes in Computer Science, vol. 9256, pp. 715-726, 2015.

8. Tomoya Kato, Hayato Itoh, Atsushi Imiya: Optical flow computation with locally quadratic assumption. *Proc. International Conference on Analysis of Images and Patterns*, Lecture Notes in Computer Science, vol. 9256, pp. 223-234, 2015.

9. Shun Inagaki, Hayato Itoh, Atsushi Imiya: Variational multiple warping for cardiac image analysis. *Proc. International Conference on Analysis of Images and Patterns*, Lecture Notes in Computer Science, vol. 9257, pp. 749-759, 2015.

10. Hayato Itoh, Atsushi Imiya, Tomoya Sakai: Global Volumetric image registration using local linear property of image manifold. *Proc. ACCV Workshop on Big data in 3D Computer Vision*, Lecture Notes in Computer Science, vol. 9008, 238-253, 2015.

11. Shun Inagaki, Hayato Itoh, Atsushi Imiya: Multiple alignment of spatiotemporal deformable objects for the average-organ computation. *Proc. ECCV Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment*, Lecture Notes in Computer Science, vol. 8928, pp 353-366, 2015.

12. Hayato Itoh, Atsushi Imiya, Tomoya Sakai: Two-dimensional global image registration using local linear property of image manifold. *Proc. 22nd International Conference on Pattern Recognition*, pp. 3862-3867, 2014.

13. Hayato Itoh, Shun Inagaki, Ming-Ying Fan, Atsushi Imiya, Kazuhiko Kawamoto, Tomoya Sakai: Local affine optical flow computation. *Proc. PSIVT Workshop on Geometric Computation for Computer Vision*, Lecture Notes in Computer Science, vol. 8334, pp 203-215, 2014.

14. Hayato Itoh, Tomoya Sakai, Kazuhiko Kawamoto, Atsushi Imiya: Global image registration using random projection and local linear method. *Proc. 15th International Conference on Computer Analysis of Images and Patterns*, Lecture Notes in Computer Science, vol. 8047, pp 564-571, 2013.

15. Hayato Itoh, Tomoya Sakai, Kazuhiko Kawamoto, Atsushi Imiya: Topology-preserving dimension-reduction methods for image pattern recognition. *Proc. 18th Scandinavian Conference on Image Analysis*, Lecture Notes in Computer Science, vol. 7944, pp. 195-204 , 2013.

16. Hayato Itoh, Tomoya Sakai, Kazuhiko Kawamoto, Atsushi Imiya: Dimension reduction methods for image pattern recognition. *Proc. Second International Workshop on Similarity-Based Pattern Analysis and Recognition*, Lecture Notes in Computer Science, vol. 7953 pp. 26-42, 2 2013.

17. Hayato Itoh, Shuang Lu, Tomoya Sakai, Atsushi Imiya: Interpolation of reference images in sparse dictionary for global image registration. *Proc. 8th International Symposium on Visual Computing*, Lecture Notes in Computer Science, vol. 7432 pp. 657-667, 2012.

18. Masaki Narita, Atsushi Imiya, Hayato Itoh: Edge detection and smoothing-filter of volumetric data. *Proc. 8th International Symposium on Visual Computing*, Lecture Notes in Computer Science, vol. 7432, pp. 489-498, 2012.

19. Tomoya Sakai, Haruhiko Nishiguchi, Hayato Itoh, Atsushi Imiya: Bifurcation of segment edge curves in scale space. *Proc. 3rd International Conference on Scale Space and Variational Methods in Computer Vision*, Lecture Notes in Computer Science, vol. 6667, pp. 302-313, 2012.

20. Hayato Itoh, Shuang Lu, Tomoya. Sakai, Atsushi Imiya: Global image registration by fast random projection. *Proc. 7th International Symposium on Visual Computing*, Lecture Notes in Computer Science, Lecture Notes in Computer Science, vol. 6938, pp. 23-32, 2011.

21. Tomoya Sakai, Hayato Itoh, Atsushi Imiya: Multi-label classification for image annotation via sparse similarity voting. *Proc. 3rd International Workshop on Subspace Methods*, Lecture Notes in Computer Science, vol. 6469(2), pp. 344-353, 2011.

Technical Reports

1. Hayato Itoh, Atsushi Imiya, Tomoya Sakai: Object oriented data analysis for volumetric medical data using multiway principal component analysis. *Technical Report of IEICE*, in Print, 2016 (English).

2. Hayato Itoh, Atsushi Imiya, Tomoya Sakai: Tensor-based methods for dimension reduction of volumetric data. *Technical Report of IEICE*, vol. 115, no. 517, PRMU2015-197, pp.197-202, 2016 (English).

3. Atsushi Imiya and Hayato Itoh:Eigenfunctions in linear scale space. *Technical Report of IEICE*, vol. 115, no. 388, pp. 87-92, PRMU2015-107, 2015 (English).

4. Hayato Itoh, Atsushi Imiya, Tomoya Sakai: Mathematical properties of the gradient-based discriminative methods. *Technical Report of IEICE*, vol. 115, no. 98, PRMU2015-50, pp.107-111, 2015 (English).

5. Hayato Itoh, Atsushi Imiya, Tomoya Sakai: Second-order tensor principal component analysis meets two-dimensional singular value decomposition. *Technical Report of IEICE*, vol. 115, no. 24, PRMU2015-14, pp.71-74, 2015 (English).

6. Hayato Itoh, Atsushi Imiya, Tomoya Sakai: 3D global image registration using local linear property in sparse dictionary. *Technical Report of IEICE*, vol. 113, no. 402, PRMU2013-102, pp.125-130, 2014 (English).

7. Hayato Itoh, Atsushi Imiya, Tomoya Sakai: Explicit local linear method for 2D affine image registration. *Technical Report of IEICE*, vol. 113, no. 346, PRMU2013-82, pp. 85-90, 2013 (English).

8. Hayato Itoh, Tomoya Sakai, Atsushi Imiya: Validation of dimension reduction methods for two-dimensional image pattern classification. *Technical Report of IEICE*, vol. 112, no. 495, PRMU2012-183, pp. 19-24, 2013 (English).

9. Hayato Itoh, Tomoya Sakai, Atsushi Imiya: Effects of dimension reduction on appearance-based pattern classification. *Technical Report of IEICE*, vol. 112, no. 357, PRMU2012-74, pp. 25-30, 2012 (English).

10. Hayato Itoh, Tomoya Sakai, Atsushi Imiya: NN-based local subspace method for image registration with sparse dictionary. *Technical Report of IEICE*, vol. 112, no. 197, PRMU2012-48, pp. 179-184, 2012 (English).

11. Hayato Itoh, Shuang Lu, Tomoya Sakai, Atsushi Imiya: Efficient global image registration using local linearity of transformed image. *Technical Report of IEICE*, vol. 111, no. 331, MI2011-68, pp. 31-36, 2011(Japanese).

12. Shuang Lu, Hayato Itoh, Tomoya Sakai, Atsushi Imiya: Global image registration using random projection. *Technical Report of IPSJ*, vol. 2011-CVIM-177, no.7, pp. 1-7, 2011 (English).

13. Hayato Itoh, Tomoya Sakai, Atsushi Imiya: Pattern classification by sparse subspace method. *Technical Report of IPSJ*, vol. 2010-CVIM-172, no.33, pp. 1-7, 2010 (Japanese).