Human gait recognition under changes of walking conditions

Author:
Kusakunniran, Worapan

Publication Date:
2013

DOI:
https://doi.org/10.26190/unsworks/16137

Human Gait Recognition under Changes of Walking Conditions

Worapan Kusakunniran

School of Computer Science and Engineering

The University of New South Wales

A thesis submitted in fulfilment
of the requirements for the degree of

Doctor of Philosophy

June 2013

PLEASE TYPE

THE UNIVERSITY OF NEW SOUTH WALES
Thesis/Dissertation Sheet

Surname or Family name: Kusakunniran

First name: Worapan

Other name/s:

Abbreviation for degree as given in the University calendar: PhD

School: Computer Science and Engineering

Faculty: Engineering

Title: Human gait recognition under changes of walking conditions

Abstract 350 words maximum: (PLEASE TYPE)

The study of human gait is innate to human interest and pervades many fields including biometrics, clinical analysis, computer animation, and robotics. From a surveillance perspective, gait recognition is capable of identifying humans at a distance by inspecting their walking manners. It is an attractive modality which can be performed surreptitiously in an unconstrained environment. Gait is one of the few biometric features that can be measured remotely without physical contact and proximal sensing, which makes it useful in surveillance applications. However, in the real world, there are various factors significantly affecting human gait including clothes, shoes, carrying objects, walking surfaces, observed views, and walking speeds. Among these factors, changes of views and speeds have been regarded as two of the most challenging problems for gait recognition. Particularly, view change will significantly impact on available visual features for matching, while speed change will alter walking patterns of each individual substantially. This thesis is mainly to develop novel methods for recognizing gaits under changes of walking conditions focusing on views and speeds, without a cooperative camera system. Five major methods are proposed from several various perspectives to address key aspects of these problems. Principally, a view-normalization of gaits is obtained through a new learning process by using mapping/projection relationships between correlated gait features across different views, while a novel speed-invariant gait feature is developed by using a statistical shape analysis based on a local-static gait information. Based on widely adopted gait databases, the comprehensive experiments are carried out to verify the proposed methods. It is concluded that the proposed methods can achieve state-of-the-art performances for gait recognitions under view change and/or speed change. In this thesis, the other relevant problems are also sorted out, including gait period analysis, view classification, and walking speed estimation. Moreover, in order to enhance the performance, multi-view gait information is utilised to achieve more stable and convincing outcomes.

Worapan Kusakunniran .... Thitinan Tantidham .... 31/05/2013 ....

Signature | Witness | Date

FOR OFFICE USE ONLY

Date of completion of requirements for Award:

THIS SHEET IS TO BE GLUED TO THE INSIDE FRONT COVER OF THE THESIS

ORIGINALITY STATEMENT

'I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.'

Signed Worapan Kusakunniran ……………………………………………………

Date 31/05/2013 ……………………………………………………

COPYRIGHT STATEMENT

AUTHENTICITY STATEMENT

I would like to dedicate this thesis to my parents for their love, patience, understanding and never-ending support.

Acknowledgements

First and foremost I want to express my deepest gratitude to my supervisor, A/Prof. Jian Zhang, and co-supervisor, Dr. Qiang Wu, for supporting me during these past four years. Dr. Zhang has always been a great supervisor. He introduced me to research, and taught me how to think intellectually and see things in different perspectives. He has instilled in me skills that will be invaluable for my future research career and private life. I will be forever grateful for his continued support and expertise. I am also thankful for the excellent example he has provided as a successful researcher.

I am very grateful to Dr. Wu for his great advices and insightful discussions. He always helped to smooth out my thoughts and ideas. I successfully overcame many difficulties and learned a lot under his supervision. I am very thankful for all his contributions of time, ideas, invaluable suggestions, and constant encouragement to make my Ph.D. experience productive and stimulating. His thoughtful guidance, enthusiasm and passion for research have always been my source of inspiration.

I am also greatful to Dr. Hongdong Li and Prof. Liang Wang for their helpful suggestions and comments. I would like to thank Prof. Yi Ma for his advices and intelligent discussions that motivated me to make a difference. I also take this chance to thank Prof. Shin'ichi Satoh for giving me an opportunity to do my internship study at National Institute of Informatics (NII) and his kind support during my stay.

I would like to thank the members of my Ph.D. committee, Dr. Alan Blair, Dr. Wei Wang, and Dr. Peter Cai for their valuable time and comments. Their great advices helped to keep my research in the right track. My thanks also go to the administrative staff of NICTA, UNSW, and NII for their outstanding assistance and support.

I extend heartfelt thanks to my colleagues at NICTA, UNSW, and NII for a lively and inspiring atmosphere. They have been a source of friendships as well as good advice and collaboration. I owe my warm thanks to all my friends from past and present, who have always been there for me when I am down. They have always cheered me up and stood by me through good times and bad times. I would like to thank them all from the bottom of my heart.

Last, but not least, I would like to express my deepest gratitude and thanks to my family for their unconditional love and infinite support throughout my life. Thank you for always loving me and believing in me. Thank you to my beloved mom and dad for being the best parents in every way. Their love, caring, and understanding mean everything to me. Finally, I would say thanks to all who have been supportive during this period of my Ph.D. study.

Abstract

In real applications, changes of observed views and walking speeds cause significant difficulties for gait recognition. View change will significantly alter available visual features for matching, while speed change will significantly alter walking patterns of each individual. This thesis is to develop novel methods for recognizing gaits under view and speed changes. Several methods are proposed from various perspectives to address diverse aspects of these problems.

Our first method is proposed to enhance a well-known gait feature, called *Gait Energy Image (GEI)*, for a robust gait recognition under changes of various walking conditions. A new *Weighted Binary Pattern (WBP)* is constructed as a robust gait feature based on partial ideas of *Local Binary Pattern (LBP)* on *GEI*. An adaptive weighting technique is also introduced to discriminate significances of each binary bit in *WBP*. Although this proposed method can handle changes of various walking conditions, its performance drops largely when view change or speed change becomes significant. Our later methods in this thesis are proposed to directly target such challenges.

Accordingly, the second method is proposed to address the problem of significant view change, based on *View Transformation Model (VTM)*. In principle, *VTM* is used to transform gait from one view into another view. Thus, gaits from different views can be normalized into a common view using learned *VTM*(s). Two types of *VTM* are proposed in this study, namely 1) *decomposition-based VTM (dVTM)*; 2) *regression-based VTM (rVTM)*. *dVTM* is established through a matrix factorization process by adopting *Singular Value Decomposition (SVD)*. The gait matrix of the training dataset is decomposed into

a view-independent matrix and a subject-independent matrix which is used to construct *dVTM*s. However, the performance of *dVTM* is bounded by limitations of linear processes on global gait features. To further improve the performance, *rVTM* reformulates the *VTM*-framework as a regression problem. It consists of multiple regression processes to model local correlated motions of gaits across views. Among several regression tools adopted for *rVTM*, *Sparse Regression (SR)* achieves the best performance. Based on our experiments, *rVTM* is shown to outperform *dVTM* and other existing methods in the literature. However, its regression framework is based on mappings to single pixels, which can be unstable especially under large view change.

To overcome the above limitations of the *VTM*-based methods, the third method is proposed based on more relaxing mappings between segments (i.e. groups of pixels) through a correlated motion co-clustering method. It follows the solution track of motion correlation analysis and explicitly takes into account inconsistencies among different parts of individual gaits across views. To this end, for the first time a *bipartite graph multipartitioning* is applied to co-cluster correlated gait segments from different views. *Canonical Correlation Analysis (CCA)* is then applied to further maximize linear correlation of the co-clustered segments. Finally, a *linear approximation* is built up to bridge these optimized gait segments across views. In this way, it is possible to normalize gaits from different views into a common *CCA* subspace where gait similarity can be measured accordingly.

The above proposed methods achieve outstanding performances on gait recognition under view change. However, they rely on supervised learning, which can be inefficient/incapable for untrained/unseen views. To avoid this restriction, the fourth method is developed from a totally different perspective to construct a new view-invariant gait feature without supervised learning. A view-normalization is processed in an input layer (i.e. on gait silhouettes). That is, each sequence of gait silhouettes recorded from a certain view is transformed onto the common canonical view by using corresponding domain transformation obtained through *Invariant Low-rank Textures (TILT)*. Then, an improved scheme of *Procrustes Shape Analysis (PSA)* is proposed and applied on a sequence of the normalized gait silhouettes to extract a novel view-invariant gait feature and consecutively measure a gait similarity.

The other method is proposed in this thesis based on a statistical shape analysis for gait recognition under speed change. A *PSA* scheme is adopted for gait signature description and relevant similarity measurement. In order to address challenges raised by speed changes, a *Higher-order Shape Configuration (HSC)* is proposed to replace the traditional shape descriptor. Moreover, instead of simply measuring a similarity between two gaits by treating them as two unified objects, a *Differential Composition Model (DCM)* is constructed to differentiate impacts caused by speed changes on various body parts.

The proposed methods are comprehensively tested on a variety of widely adopted gait databases. They are shown to achieve state-of-the-art performances for gait recognitions under both view change and speed change.

List of Publications

Journals

- W. Kusakunniran, Q. Wu, J. Zhang, and H. Li. Cross-view and multi-view gait recognitions based on view transformation model using multi-layer perceptron. Pattern Recognition Letters (PRL), 33(7):882-889, May 2012.
- W. Kusakunniran, Q. Wu, J. Zhang, and H. Li. Gait recognition under various viewing angles based on correlated motion regression. IEEE Tran. on Circuits and Systems for Video Technology (TCSVT), 22(6):966–980, February 2012.
- W. Kusakunniran, Q. Wu, J. Zhang, and H. Li. Gait recognition across various walking speeds using higher-order shape configuration based on a differential composition model. IEEE Tran. on Systems, Man, and Cybernetics, Part B: Cybernetics (TSMCB), 42(6):1654–1668, May 2012.
- W. Kusakunniran, Q. Wu, J. Zhang, Y. Ma, and H. Li. A new view-invariant feature for cross-view gait recognition. IEEE Tran. on Information Forensics and Security (TIFS), March 2013.

Referred conference papers

- W. Kusakunniran, Q. Wu, H. Li, and J. Zhang. Automatic gait recognition using weighted binary pattern on video. pages 49–54, Italy, September 2009. IEEE Conf. on Advanced Video and Signal Based Surveillance (AVSS).

- W. Kusakunniran, Q. Wu, H. Li, and J. Zhang. Multiple views gait recognition using view transformation model based on optimized gait energy image. pages 1058–1064, Japan, September-October 2009. IEEE Int. Conf. on Computer Vision (THEMIS workshop in conjunction with ICCV).
- W. Kusakunniran, H. Li, and J. Zhang. A direct method to self-calibrate a surveillance camera by observing a walking pedestrian. pages 250–255, Australia, December 2009. Digital Image Computing: Techniques and Applications (DICTA).
- W. Kusakunniran, Q. Wu, J. Zhang, and H. Li. Support vector regression for multi-view gait recognition based on local motion feature selection. pages 974–981, United States, June 2010. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR).
- W. Kusakunniran, Q. Wu, J. Zhang, and H. Li. Multi-view gait recognition based on motion regression using multilayer perceptron. pages 2186–2189, Turkey, August 2010. Int. Conf. on Pattern Recognition (ICPR).
- W. Kusakunniran, Q. Wu, J. Zhang, and H. Li. Speed-invariant gait recognition based on procrustes shape analysis using higher-order shape configuration. pages 545–548, Belguim, September 2011. IEEE Int. Conf. on Image Processing (ICIP).
- W. Kusakunniran, Q. Wu, J. Zhang, and H. Li. Pairwise shape configuration-based PSA for gait recognition under small viewing angle change. pages 17–22, Austria, August 2011. IEEE Conf. on Advanced Video and Signal Based Surveillance (AVSS).
- W. Kusakunniran, S. Satoh, J. Zhang, and Q. Wu. Attribute-based learning for large scale object classification. United States, July 2013. IEEE Int. Conf. on Multimedia and Expo (ICME).

Awards

- Best Biometrics Student Paper Award (BBSPA) in Int. Conf. on Pattern Recognition (ICPR) 2010.

Submitted papers

- W. Kusakunniran, Q. Wu, J. Zhang, H. Li, and L. Wang. Recognizing gaits across views through correlated motion co-clustering. IEEE Tran. on Image Processing (TIP).

Contents

1 Introduction **1**

1.1 Overall Problem Definitions and Research Challenges 2

1.1.1 Gait recognition under view change 2

1.1.2 Gait recognition under speed change 4

1.2 Related Works 6

1.2.1 Gait recognition under no change of walking condition (i.e. fixed view and speed) 6

1.2.2 Gait recognition under view change 8

1.2.3 Gait recognition under speed change 12

1.3 Gait Databases 14

1.3.1 The CASIA gait database A 14

1.3.2 The CASIA gait database B 14

1.3.3 The CASIA gait database C 15

1.3.4 The USF gait database 15

1.3.5 The OU-ISIR gait database 16

1.3.6 The CMU Mobo gait database 16

1.3.7 Overall comparisons on various gait databases 16

1.4 Contributions 16

1.5 Thesis Outline 22

2 Fundamental Knowledge **23**

2.1 Gait Energy Image (GEI) 23

2.2 Local Binary Pattern (LBP) 25

2.3 Subspace Analysis 26

2.3.1 Principal Component Analysis (PCA) 26

2.3.2 Linear Discriminant Analysis (LDA) 27
2.4 Singular Value Decomposition (SVD) 27
2.5 Regression Tools 28
2.5.1 Multi-Layer Perceptron (MLP) 29
2.5.2 Support Vector Regression (SVR) 30
2.5.3 Sparse Regression (SR) 33
2.6 Bipartite Graph Partitioning 35
2.6.1 Bipartite graph bipartitioning via SVD 36
2.6.2 Bipartite graph multipartitioning via SVD 37
2.7 Canonical Correlation Analysis (CCA) 38
2.8 Procrustes Shape Analysis (PSA) 39
2.9 Transform Invariant Low-rank Textures 41

3 Automatic Gait Recognition using Weighted Binary Pattern on Video 44
3.1 Introduction 44
3.2 Pre-processing and Gait Period Analysis 45
3.3 Weighted Binary Pattern (WBP) 48
3.3.1 Partial local binary pattern based features 48
3.3.2 Variance bit-weighting analysis 49
3.3.3 Similarity measurement 51
3.4 Experiments 51
3.4.1 Results on the CASIA gait database A 52
3.4.2 Results on the CMU Mobo gait database 53
3.4.3 Results on the CASIA gait database C 55
3.5 Conclusion 57

4 Gait Recognition under Various Views based on View Transformation Model 59
4.1 Introduction 59
4.2 Problem Formulation 61
4.3 Pre-processing and View Classification 65
4.3.1 LDA in PCA transformed space 66
4.3.2 View classifier 66

4.4 Gait Feature Preparation 67
4.5 Solution based on dVTM 69
4.6 Solution based on rVTM 71
4.6.1 ROI selection 72
4.6.2 Regression training process 76
4.7 Gait similarity measurement 80
4.8 Typical Implementations for Different Cases of Gait Recognition under Various Views 81
4.8.1 Cross-view gait recognition based on a one-view to one-view VTM 81
4.8.2 Cross-view gait recognition based on one-view to one-view VTMs through middle view 82
4.8.3 Multi-view gait recognition based on a multi-view to one-view VTM 82
4.8.4 Optimal view selection for multi-view gait recognition . . . 83
4.9 Experiments 84
4.9.1 View classification 85
4.9.2 Performance evaluation on ROI selection 87
4.9.3 Cross-view gait recognition based on a one-view to one-view VTM 89
4.9.4 Cross-view gait recognition by transforming gaits under source and target views into another common view 91
4.9.5 Multi-view gait recognition based on a multi-view to one-view VTM 92
4.9.6 View selection for multi-view gait recognition 93
4.9.7 Cross-view gait recognition under various conditions 95
4.9.8 Cross-view gait recognition under outdoor environment . . 95
4.10 Discussion and Conclusion 96

5 Gait Recognition under Various Views based on Correlated Motion Co-clustering 99
5.1 Introduction 99
5.2 Gait Feature Preparation 102

5.3 Gait Partitioning Model 104
5.3.1 Problem formulation based on co-clustering 104
5.3.2 Bipartite graph modelling on gait features from different views 105
5.3.3 Bipartite graph multipartitioning on gait features from different views 106
5.4 View-Normalization based on Correlation Optimization and Linear Approximation 108
5.4.1 Correlation optimization 108
5.4.2 Linear approximation 109
5.5 Gait Recognition under Various Views 109
5.5.1 Cross-view gait recognition 110
5.5.2 Multi-view gait recognition 110
5.6 Experiments 111
5.6.1 GEI partitioning model 112
5.6.2 Cross-view gait recognition 113
5.6.3 Multi-view gait recognition 114
5.6.4 Cross-view gait recognition in a cluttered outdoor environment 115
5.7 Conclusion 116

6 Gait Recognition Across Various Walking Speeds Using Higher Order Shape Configuration 120

6.1 Introduction 120
6.2 Pre-processing and Walking Speed Estimation 123
6.3 PSA-based Gait Recognition 126
6.3.1 CSC-based shape descriptor and its constraint to walking speed change 127
6.3.2 PMS-based gait feature 127
6.3.3 PD-based gait similarity measurement and its constraint to walking speed change 128
6.4 Impacts of Speed Change on PSA-based Gait Recognition 129

6.4.1 The change of walking speed vs. the change of walking pattern 130
6.4.2 The change of walking speed vs. the higher-order derivatives of the shape boundary 132
6.4.3 The change of walking speed vs. the impacts on the different body parts 134
6.4.4 The change of walking speed vs. the response of the individual person 136
6.5 Speed-invariant Gait Recognition based on Improved PSA 138
6.5.1 Shape descriptor based on HSC 139
6.5.2 DCM 142
6.6 Experiments 147
6.6.1 Results on the CASIA gait database C 148
6.6.2 Results on the OU-ISIR gait database 151
6.6.3 Results on the CMU Mobo gait database 153
6.6.4 Results on the USF gait database 154
6.7 Conclusion 156

7 View-invariant Gait Recognition based on Low-rank Textures Analysis 157
7.1 Introduction 157
7.1.1 Rationale of the proposed method 158
7.1.2 Summary of contributions and advantages 160
7.2 Framework of the Proposed Solution 161
7.3 View-normalization 163
7.3.1 Gait Texture Image 163
7.3.2 TILT for view-normalization 165
7.4 Gait Feature Extraction 168
7.4.1 Shape descriptor 169
7.4.2 PMS-based gait feature 170
7.5 Gait Similarity Measurement 172
7.5.1 Constraint to view change 172
7.5.2 PD-based gait similarity measurement 173

7.6 Experiments 174
7.6.1 Comparison with other methods using view-invariant gait feature (i.e. the second category) 175
7.6.2 Comparison with other methods relying on learning mapping/projection relationship of gaits across views (i.e. the third category) 176
7.6.3 Robustness when increasing a number of subjects in the dataset 176
7.6.4 Cross-view gait recognition under outdoor environment . . 177
7.7 Conclusion 178

8 Conclusions and Future Works **182**
8.1 Summary of Research 182
8.2 Future Works 185
8.2.1 Combination of the proposed methods for more practical situations 186
8.2.2 Combination of gait with other biometrics 188
8.2.3 Gait analysis for other research fields 189

References **213**

List of Figures

1.1 View change under two different cases. 2
1.2 Sample gait images under various views from the CASIA gait database B. 3
1.3 Sample gait images under various speeds from the OU-ISIR gait database. 5
1.4 Sample gait images under various views from the CASIA gait database A. 14
1.5 Sample gait images under various walking conditions from the CASIA gait database C. 14
1.6 Sample gait images under the left and right cameras from the USF gait database. 15
1.7 Sample gait images under various walking conditions from the CMU Mobo gait database. 15

2.1 Sample *GEI*s under various views, clothes, and carrying conditions. 24
2.2 The basic *LBP* operator. 25
2.3 Examples of multi-scale *LBP* operator. 25
2.4 Example of the typical neural network with one hidden layer and one output node. 30
2.5 The soft margin lose settings for a linear SVM and a nonlinear SVM. 31

3.1 *Gait period* analysis. 47
3.2 CMS curves on the CASIA gait database C for the case of fixed walking condition. 56

3.3 CMS curves on the CASIA gait database C for the case of across walking conditions. 57

4.1 The proposed framework of *VTM*-based gait recognition. 64
4.2 Correlated motions between *GEI*s from different views. 68
4.3 *ROI* selection. 73
4.4 Projection histograms for body part segmentation process. 74
4.5 Relationship between source *ROI* and its corresponding target pixel. 75
4.6 Examples of *ROI*s. 80
4.7 Examples of transformed *GEI*s. 85
4.8 Comparisons on cross-view gait recognition (%) using the proposed *rVTM*-based methods without *ROI* selection (*GEI-SVR*, *GEI-SR*) and the proposed *rVTM*-based methods with *ROI* slelection (*ROI-SVR*, *ROI-SR*). 87
4.9 Comparisons on cross-view gait recognition (%) using different *VTM*-based methods including FT-*dVTM*, *GEI-dVTM*, *MLP-rVTM*, *SVR-rVTM*, and *SR-rVTM*. 88
4.10 Comparisons on cross-view gait recognition based on an one-view to one-view *VTM* (%) using different methods. 91
4.11 Comparisons on *cross-view gait recognition* based on one-view to one-view *VTM*s through middle view (%) using different methods. 92
4.12 Comparisons on *multi-view gait recognition* based on a multi-view to one-view *VTM* (%) using different methods. 93

5.1 The proposed framework for *cross-view gait recognition*. 100
5.2 The correlations between *GEI*s from different views. 102
5.3 Graph partitioning. 106
5.4 Sample results of *GEI* partitioning model. 112

6.1 The framework of *PSA*-based gait recognition. 126
6.2 Examples of PMS-based gait feature from various speeds. 129
6.3 Boundary point variations between different speeds. 130
6.4 Boundary segmentation. 131

6.5 Efforts caused by the walking speed change on the variance of Euclidean distance, and its 1^{st} order (tangent), 2^{nd} order (curvature) and 3^{rd} order (aberrancy) derivatives respectively at each boundary point. 133
6.6 Patterns of shape change caused by speed change and view change. 136
6.7 The speed-invariant *PSA*-based gait recognition. 138
6.8 Information entropy representing quantity of gait information. . . 140
6.9 The weight analyses using different type of shape descriptors. . . . 146
6.10 The weight analyses on different boundary segments corresponding to each body part. 146

7.1 The proposed framework of view-invariant gait recognition. 161
7.2 The proposed framework of view-normalization. 163
7.3 *GTI*s under various views. 164
7.4 *Nuclear norm* of *GTI* from various views. 164
7.5 Sample results from *TILT* optimization. 166
7.6 Sample transformed gait silhouettes from various views into the common canonical view. 167
7.7 The difference between two silhouettes from two different views. . 168
7.8 The proposed framework of gait feature extraction. 169
7.9 Example of PMS-based gait features on the CASIA gait database B. 171
7.10 Example of PMS-based gait features of the same subject from different views on the CASIA gait database B. 171
7.11 Rotation adjustment between PMS-based gait features of the same subject from different views. 173
7.12 The performance of *cross-view gait recognition* using the proposed method against increasing the number of subjects in the dataset. . 177

8.1 Combination of the proposed methods of gait recognition under view change. 187

List of Tables

1.1 Comparisons between different categories of methods for gait recognition under view change. 12
1.2 Comparisons between different adopted gait databases. 17

3.1 Experimental results on the CASIA gait database A. 52
3.2 Comparison of rank 1 performance on the CASIA gait database A. 53
3.3 Experimental results on the CMU Mobo gait database for the case of fixed walking condition. 54
3.4 Experimental results on the CMU Mobo gait database for the case of across walking conditions. 54
3.5 Comparison of rank 1 performance on the CMU Mobo gait database. 55
3.6 Twelve experiments on the CASIA gait database C for the case of across walking conditions. 55
3.7 Comparison of rank 1 performance on the CASIA gait database C. 58

4.1 Rank 1 *view classification* results (%) for each view from the CASIA gait database B. 86
4.2 Influence of automatic *view classification* on the overall performance of cross-view gait recognition. 86
4.3 Evaluation of *multi-view gait recognition* (%) using the proposed method. 93
4.4 Comparisons on *multi-view gait recognition* (%) using different methods. 94
4.5 Performance of *cross-view gait recognition* (%) under various walking conditions. 95

4.6 Performance of *cross-view gait recognition* (%) under outdoor environment using the experiment A of the USF gait database. . . . 96

5.1 Table of notations related to gait features. 103
5.2 Comparisons on *cross-view gait recognition* (%) using different methods. 117
5.3 Comparisons on *multi-view to one-view gait recognition* (%) using different methods. 118
5.4 Comparisons on *multi-view to multi-view gait recognition* (%) using different methods. 119
5.5 Performance of cross-view gait recognition (%) in a cluttered outdoor environment using the experiment A of the USF gait database. 119

6.1 Estimation of *gait period* for the CASIA gait database C. 124
6.2 Estimation of *gait period* for the OU-ISIR gait database. 125
6.3 Gait shape change (based on the measurement with PD) caused by speed change. 135
6.4 Gait recognition performance (%) on the CASIA gait database C using the proposed method based on different types of shape configuration and PD without *DCM*. 148
6.5 Gait recognition performance (%) on the CASIA gait database C using the proposed method based on different types of shape configuration and PD with *DCM*. 149
6.6 Comparison of cross-speed gait recognition performance (%) on the CASIA gait database C using different methods. 150
6.7 Gait recognition performance (%) on the OU-ISIR gait database using the proposed method without *DCM*. 151
6.8 Gait recognition performance (%) on the OU-ISIR gait database using the proposed method with *DCM*. 152
6.9 Comparison of cross-speed gait recognition performance (%) on the OU-ISIR gait database using the proposed method between with and without *DCM*. 153
6.10 Comparison of cross-speed gait recognition performance (%) using different methods. 154

6.11 Comparison of cross-speed gait recognition performance (%) on the CMU Mobo gait database using different methods. 155
6.12 Comparison of gait recognition performance (%) on the experiment A of the USF gait database using different methods. 156

7.1 Comparison on *cross-view gait recognition* (%) using different methods in the second category 179
7.2 Comparison on *cross-view gait recognition* (%) using different methods in the third category. 180
7.3 Comparison on *cross-view gait recognition* (%) in the experiment A of the USF gait database using different methods. 181

Index

Page numbers indicate the primary sources of information for the corresponding topics.

bipartite graph bipartitioning, 36
bipartite graph multipartitioning, 37
canonical correlation analysis, 38
CCA, see canonical correlation analysis
centroid shape configuration, 127
cross-view gait recognition, 3, 81, 110
CSC, see centroid shape configuration
DCM, see differential composition model
decomposition-based VTM, 69
differential composition model, 142
dVTM, see decomposition-based VTM
Fisher discriminant ratio, 144
fixed-view gait recognition, 2
gait energy image, 23
gait partitioning, 104
gait period, 45
gait texture image, 163
GEI, see gait energy image
GTI, see gait texture image
higher-order shape configuration, 139
HSC, see higher-order shape configuration
LBP, see local binary pattern
LDA, see linear discriminant analysis
linear approximation, 109

linear discriminant analysis, 27
local binary pattern, 25
MLP, see multi-layer perceptron
multi-layer perceptron, 29
multi-view gait recognition, 3, 82, 83, 110
multi-view to multi-view gait recognition, 3, 110
multi-view to one-view gait recognition, 3, 82, 83, 110
nuclear norm, 43
pairwise shape configuration, 169
PCA, see principal component analysis
principal component analysis, 26
Procrustes shape analysis, 39
PSA, see Procrustes shape analysis
PSC, see pairwise shape configuration
region of interest, 72
regression-based VTM, 71
ROI, see region of interest
rVTM, see regression-based VTM
singular value decomposition, 27
sparse regression, 33
SR, see sparse regression
support vector regression, 30
SVD, see singular value decomposition
SVR, see support vector regression
TILT, see transform invariant low-rank textures
transform invariant low-rank textures, 41
view classification, 65
view transformation model, 59
VTM, see view transformation model
walking cycle, 45
walking speed estimation, 124
WBP, see weighted binary pattern

weighted binary pattern, 48

Chapter 1

Introduction

The study of human gait is innate to human interest and pervades many fields including biometrics, clinical analysis, computer animation and robotics [1]. From a surveillance perspective, gait recognition is capable of identifying humans at a distance by inspecting their walking manners. It is an attractive modality which can be performed surreptitiously in an unconstrained environment. Gait is one of the few biometric features that can be measured remotely without physical contact and proximal sensing, which makes it useful in surveillance applications.

In the real world, there are various factors significantly affecting human gait including clothes, shoes, carrying objects, walking surfaces, observed views, and walking speeds [2]. Among these factors, view change and speed change have been regarded as two of the most common and challenging problems for gait recognition.

In particular, view change will significantly alter available visual features for matching, while speed change will significantly alter walking patterns of each individual. Thus, gaits from different views and/or speeds must be normalized before their similarity is measured. These problems will be addressed in this study.

The rest of this chapter is organized as follows. Section 1.1 begins with an introduction to the problems and discussions of the challenges. Section 1.2 gives literature reviews on related works. Section 1.3 introduces gait databases adopted for our experiments. Section 1.4 summarizes our contributions. Section 1.5 concludes with an outline of this thesis.

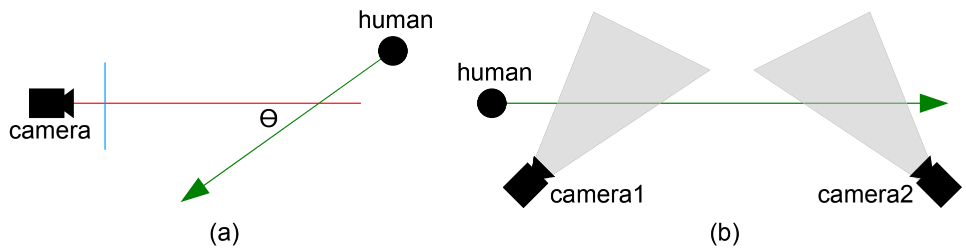


Figure 1.1: View change under two different cases. (a) Humans are free to walk in any direction covered by a single camera. (b) Humans can walk through multiple cameras. The blue line is an image plane. The red line is an optical axis. The green arrows are walking directions. The grey polygons are areas covered by cameras. The θ is an observed view which is an angle between an optical axis and a walking direction.

1.1 Overall Problem Definitions and Research Challenges

As mentioned above, the main objective of this thesis is to address the problems of view change and speed change for gait recognition. This section will give the problem definitions and expose the relevant challenges. Then, our motivations to approach such problems will be explained.

1.1.1 Gait recognition under view change

1.1.1.1 Problem definition

In practice, observed view may change under two different cases. The first case is shown in Figure 1.1(a) that a person may walk freely in any direction facing a camera. The second case is shown in Figure 1.1(b) that a person may walk across multiple cameras which have different settings such as the height and rotation of the camera. Sample gait images under various views from the CASIA gait database B [3] are shown in Figure 1.2.

As mentioned above, gaits can be recorded from different views. Consequently, probe gait can possibly be captured under an arbitrary view which does not match any view in the gallery dataset. In this study, three scenarios are defined for gait recognition regarding view factor, namely 1) *fixed-view gait recognition* where

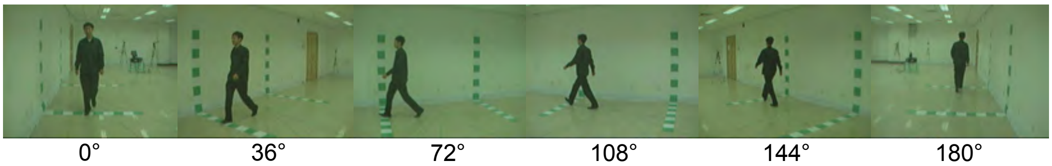


Figure 1.2: Sample gait images under various views from the CASIA gait database B.

probe gait and gallery gait are recorded from the same view; 2) *cross-view gait recognition* where probe gait and gallery gait are recorded from two different views; 3) *multi-view gait recognition* where probe gait(s) from single/multiple view(s) is recognized by using gallery gaits from multiple views. Moreover, the third scenario is also called *multi-view to one-view gait recognition* when the probe gait of each individual is recorded from a single view, otherwise it is called *multi-view to multi-view gait recognition*.

View-normalization is necessary for the second and third scenarios which will be addressed in this thesis.

1.1.1.2 Challenge

Individual gaits will exhibit in dramatically different ways under various views. Changes of view can lead to the following changes of gait information.

- visible body parts
- visible motions
- global shape statistics
- body proportions
- walking trajectories
- self-occlusions

Obviously, it is not reasonable to directly measure similarity between two gaits under different views without any view-normalization process. Gait recognition performance will dramatically drop when views change [4][5]. Moreover, previous works [6][7] have proved that gait information is maximized under side view (i.e. 90°).

1.1.1.3 Motivation

Although individual gaits vary across different views, they are still related to some extent depending on the degree of view change. This is because they are captured from the same 3D human motion. In fact, they contain the following associations: 1) shared gait information; 2) correlated gait information; 3) unrelated gait information.

First, shared gait information is derived from common body parts and/or motions visible across views. However, it can be misaligned due to different geometric projections under different views. This should be rectified in a view-normalization process.

Second, correlated gait information is induced from different seen features but on correlated body parts and/or motions. For example, different parts of the thigh share correlated motion because they are physically connected and are likely to move in a unified motion. This type of information is important because it is usually a major factor in bridging gaits across views. That is, some features invisible in one view can be predicted by using correlated features visible in other views. This process will be well formulated in a view-normalization process.

Third, due to self-occlusions, some motions visible in one view disappear totally in other views. Such unrelated gait information will not contribute to gait recognition under view change. It should be eliminated in a view-normalization process.

Our proposed methods for *cross-view* and *multi-view gait recognitions* have been motivated by these rationales.

1.1.2 Gait recognition under speed change

1.1.2.1 Problem definition

Walking speed is one of the most common factors affecting gaits because humans often change their walking speed depending on situations. Although speed changes are useful for classifying gait styles (e.g. walking, running, jogging) [8],

Figure 1.3: Sample gait images under various speeds from the OU-ISIR gait database.

they are a nuisance for recognizing gaits. This is because an individual walking pattern varies across different walking speeds. Sample gait images under various speeds from the OU-ISIR gait database [9] are shown in Figure 1.3. Thus, to cope with the problem of speed changes, either gait feature transformation between different speeds or speed-invariant gait feature extraction is required.

In the absence of significant external factors, humans tend to walk at about 1.4 m/s (i.e. 5.0 km/h) [10][11][12]. Although humans are capable of walking at speeds from nearly 0 m/s to upwards of 2.5 m/s (i.e. 9.0 km/h), humans typically choose to use only a small range within these speeds [13]. Individuals find exceptionally fast or slow speeds uncomfortable. Thus, large speed changes ($\pm > 2.0$ km/h) may not be a critical problem in practice. In this way, relatively small speed changes ($\pm < 2.0$ km/h) must be seriously considered in the context of gait recognition.

1.1.2.2 Challenge

An increase in walking speed can lead to the following major impacts on human gaits [14].

- arms swing higher
- legs lift up higher
- stride length becomes longer
- *gait period* is shorter

In fact, speed change also affects other body parts including hip, knee, and ankle. Moreover, different people may react differently to the change of walking speed because of age, gender, individual body structure, etc., which makes the situation more complicated.

1.1.2.3 Motivation

When a person changes his/her walking speed, dynamic information (e.g. the movements of arms and legs) is changed while static information (e.g. thigh and shin lengths) is unchanged. In this way, speed change will have only a small impact on local shape information (e.g. tangents/curvatures around thigh and shin contours). These rationales motivate us to develop a new speed-invariant gait feature which is based on the higher-order derivatives (e.g. tangent and curvature) of shape boundary information.

1.2 Related Works

The methods of gait recognitions under no walking variation (i.e. fixed view and speed), under view change, and under speed change are reviewed in sections 1.2.1, 1.2.2, and 1.2.3 respectively.

1.2.1 Gait recognition under no change of walking condition (i.e. fixed view and speed)

A large number of gait recognition methods have been published recently, which can be roughly divided into two categories, model-based methods [15][16][17][18][19][20][21][22][23][24] and motion-based methods [25][26][27][28][29][30][31][32][33][34].

1.2.1.1 Model-based gait recognition

The model-based methods generally aim to model kinematics of human joints in order to measure physical gait parameters such as trajectories, limb lengths, and angular speeds. For example, Cunado et al. [18] considered legs as an interlinked pendulum. Then, a phase-weighted Fourier magnitude spectrum was used to recognize gait signatures which were derived from frequency components of the variations in the inclination of human thigh. Johnson and Bobick [16] used activity-specific static body parameters for gait recognition without directly analyzing dynamics of gait patterns.

Lee and Elgammal [21] introduced a framework for simultaneous gait tracking and recognition using person-dependent global shape deformations which were modeled using a nonlinear generative model with kinematic manifold embedding and kernel mapping. Then, the generative model of global shape deformation was used to estimate shape style, geometric transformation, and body pose within Bayesian framework. Bouchrika and Nixon [17] used elliptic Fourier descriptors to extract crucial features from human joints. However, methods in this category have to deal with localizations of human joints, which are not robust on a markerless motion [23]. It is also difficult to extract underlying models from gait sequences [32].

1.2.1.2 Motion-based gait recognition

The motion-based methods typically analyze gait sequences without explicit modelling of human body structures. The different methods in this category have been developed from different perspectives. For example, Abdelkader et al. [35][36] proposed an eigengait method using image self-similarity plots. Chai et al. [37] introduced a Perceptual Shape Descriptor technique for recognizing gaits. Tan et al. [30] used eight kinds of projective features to describe human gait and *PCA* was applied for gait feature dimension reduction.

Han and Bhanu [27] proposed a concept of *Gait Energy Image (GEI)*, and combined real and synthetic templates to improve the accuracy of gait recognition. Liu et al. [28] employed a population Hidden Markov Model (pHMM) to model human gaits and generated dynamics-normalized stance-frames to recognize individuals. Recently, Wang et al. [32] developed a temporal template, named Chrono-Gait Image (CGI), by encoding gait contours using a multi-channel mapping function. Roy et al. [29] modelled a gait cycle using a chain of key poses which were then averaged to generate the gait feature, called Pose Energy Image (PEI).

1.2.2 Gait recognition under view change

Current research that directly targets the problem of gait recognition under view change falls into three categories.

1.2.2.1 3D gait feature construction

Methods in the first category [38][39][40][41] are to construct 3D gait information through multiple calibrated cameras. Methods in this category belong to *multi-view gait recognition* because gaits from multiple views are acquired to reconstruct a 3D gait model. An image-based visual hull (IBVH) was invented in [39] to render visual views for gait recognition. IBVH was computed from a set of monocular views captured by multiple calibrated cameras. In this method, canonical visual camera positions were estimated. Then, rendered images obtained from these viewpoints were used for view-normalization. Bodor et al. [38] applied an image-based rendering on a 3D visual hull model to automatically reconstruct gait features under any required view. It integrated several gaits which were acquired by multiple cameras under different views. The system was required to calibrate individual cameras and to cross-calibrate them into a common ground frame.

Zhang et al. [40] introduced a view-independent gait recognition method based on a 3D linear model and Bayesian rule. The 3D linear model was constructed using *PCA* from a set of Fourier represented examples. The sets of coefficients were used as signature to describe gait, which were derived from projecting 2D gait sequences under different views onto a 3D model by means of a maximum of posterior estimate. Zhao et al. [41] reconstructed a 3D gait model from video sequences captured by multiple cameras. Motion trajectories of lower limbs that were extracted from 3D models, were used as dynamic features and linear time normalization was exploited for matching and recognition. In general 3D analysis, at least two cameras are required. However, because of self-occlusion, gaits from at least four cameras were required for sufficient 3D gait analysis [41].

There are a few limitations in this category: 1) it is only suitable for a fully controlled and cooperative multi-camera environment such as a biometric tunnel

[42]; 2) deploying a cooperative multi-camera setup into current surveillance systems is costly and complicated; 3) acquiring complex calibration information [43] for the 3D reconstruction and/or 2D rendering process may involve an expensive computation; 4) gait is a kind of non-rigid dynamic feature such that its analysis is sensitive to several factors such as occlusions and shadows, so the 3D modeling is not stable.

1.2.2.2 View-invariant gait feature extraction

The second category [44][45][46][47] is to extract gait feature which is invariant to view change. These methods are applied for *cross-view gait recognition*. It is difficult to seek a view-invariant gait feature because view-dependent information is embedded complexly in gait. The different methods in this category may be developed from completely different perspectives.

For example, Kale et al. [47] developed a method to generate a side-view of gait from any arbitrary view. Two techniques were proposed based on: 1) a perspective projection model; 2) an optical flow structure. The performance of this method significantly dropped when an angle between image plane and sagittal plane was large. Jean et al. [46] proposed a method to compute view-normalized trajectories of body parts which were obtained from monocular video sequences. The normalized feet and head 2D trajectories from tracked silhouettes were used as view-invariant gait features since they always appeared like being seen from a frontal parallel viewpoint. However, this method efficiently worked only for a limited range of views. Han et al. [45] extracted view-invariant features from *GEI*. In such a way, only parts of gait sequences that overlap between views were selected for constructing a representation of cross-view gait matching.

A key method in this category was proposed in [44] to compute a self-calibrating view-invariant gait recognition based on model-based gait features. Lower limbs' poses were estimated based on markerless motion estimation. Then, they were reconstructed in the sagittal plane using viewpoint rectification under an assumption that articulated leg motion is approximately planar. Angular measurements and trunk spatial displacements were derived from the rectified limbs' poses and used as a view-invariant gait feature. An advantage of this method is that it can efficiently perform *cross-view gait recognition* when view difference is large. However, it also has a number of limitations: 1) the limbs' poses estimation is not robust from markerless motion; 2) it is not applicable for frontal view because limbs' poses become untraceable.

1.2.2.3 Gait feature transformation between different views

The third category [48][49][50][51][52] relies on learning mapping/projection relationship of gaits across views. The relationship obtained through training will normalize gait features from different views into shared/associated subspace(s) before gait similarity is measured. Based on the literature review, methods in this category can be applied for both *cross-view* and *multi-view gait recognitions*, although this has not been mentioned in [48]. Compared with the first category, the third category uses a simpler non-cooperative camera system. The relationships between gaits from different views across cameras are determined through the learning process. Compared with the second category, the third category is more efficient and stable when sufficient training samples are supplied to the learning process.

Because of the advantages of the third category, in recent years, a number of methods have been proposed in this category. The efficient method based on a concept of *View Transformation Model (VTM)* was introduced in [52] to transform gait feature from one view into another view. *VTM* was established through a matrix factorization process by adopting *Singular Value Decomposition (SVD)*. The gait matrix in the training dataset was decomposed into a view-independent matrix and a subject-independent matrix which was used to construct *VTM*s. The method proposed in [52] created *VTM* based on a frequency-domain gait feature that obtained through Fourier Transformation.

To improve the performance of the method in [52], a *VTM* in [50] was created based on *GEI* optimized by *Linear Discriminant Analysis (LDA)*. The truncated *SVD* was also introduced to avoid oversizing and overfitting of *VTM*. However, the size of *VTM* in [50][52] depends on the size of the training dataset. To obtain a reliable *VTM*, it requires a large training dataset causing a large size of *VTM* which leads to an unpopular computational complexity. Moreover, the performance of the *decomposition-based VTM* [50][52] was bounded by limitations of linear processes on global features.

Recently, to overcome these limitations, *VTM* construction is reformulated as a regression problem [49]. The regression concept was used to reveal correlated motions of gaits from different views. *VTM* was consisted of multiple regression processes which were trained to estimate gait feature under one view using correlated information in gait feature(s) under other view(s). The local correlated features under shared motions were exposed using rough body segmentations and correlation coefficients.

Moreover, the method in [51] learned *LDA*-subspaces to extract discriminative information from gait features under each view in the training dataset. In the testing phase, each gait feature was projected onto each of the subspaces separately. Then, the final gait distance was a weighted sum of matching results from each subspace. The method in [48] is another efficient method published recently, which modelled correlations of gaits across views using *Canonical Correlation Analysis (CCA)*. Gait features from two different views were projected into other two subspaces which were maximally correlated based on *CCA*. Then, a correlation strength was applied to measure gait similarity.

A common limitation of the methods in this category is they rely on supervised learning. That is, the relationship between any two views has to be established beforehand through the training process. Consequently, it has to make sure that gaits from all interested views must be available in the training database. Due to uncertainty in a real surveillance environment, view change is not predictable. It is practically impossible to establish the training process to cover all likely views. Thus, when using methods of the third category, there are always some views where we cannot achieve on *cross-view gait recognition*.

1.2.2.4 Overall comparisons on different solutions

Comparisons are shown in Table 1.1 to summarize key differences between these three categories of methods for gait recognition under view change.

The methods in different categories contain different advantages and disadvantages. A proper method can be selected depending on a nature of each application.

Table 1.1: Comparisons between different categories of methods for gait recognition under view change.

Property	Category 1	Category 2	Category 3
cross-view gait recognition	$\times$	$\surd$	$\surd$
multi-view gait recognition	$\surd$	$\times$	$\surd$
cooperative multi-camera system i.e. costly and complicated	$\surd$	$\times$	$\times$
key joints extraction on markerless motion i.e. not robust	$\times$	$\surd$	$\times$
not applicable for approximate frontal views	$\times$	$\surd$	$\times$
not applicable for untrained/unknown views	$\times$	$\times$	$\surd$
supervised learning	$\times$	$\times$	$\surd$

1.2.3 Gait recognition under speed change

Current research on cross-speed gait recognition falls into two categories.

1.2.3.1 Speed-invariant gait feature extraction

Some interesting works in this first category are briefed below. Liu et al. [28] developed a HMM-based time-normalized gait feature. Similarity between two normalized gait features was measured using a sum of shape distances corresponding to gait stances in *LDA* space. The normalized gait dynamic based on a population-based generic walking model has shown its effectiveness to compensate the hard covariates caused by walking speed change.

Tan et al. [30] used eight kinds of projective features to describe human gait and *PCA* was applied for reduction of raw gait feature dimension. Mahalanobis distance was used to measure gait similarity. A projective normalization was used to improve the robustness of projective frieze patterns against speed variation. Kusakunniran et al. [53] applied a partial *LBP* concept on *Gait Energy Image (GEI)* and proposed adaptive weighting techniques to discriminate significant bits of partial *LBP* in gait features.

However, a typical limitation of existing methods in this category is that the performance is satisfying only to the small change of walking speed. They cannot well handle the larger change of walking speed.

1.2.3.2 Gait feature transformation between different speeds

The essential of the second category is to learn mapping relationship between gaits under different waking speeds. It transforms gait features using generic speed transformation models. Therefore, gait similarity measurement can be carried out under a common walking speed.

Tanawongsuwan et al. [54] applied knowledge learned from the stride length analysis [55]. The linear relationship between stride length and walking speed at the population/global level was used to normalize gaits across different speeds. Tsuji et al. [9] proposed a factorization-based speed transformation model using *SVD* to transform dynamic gait features from one speed to another.

The challenges to the second category are: 1) they are not applicable to unknown walking speed which is not covered by supervised training for learning the transform relationship; 2) they require model fitting and/or body part tracking; 3) they cannot achieve good performance for the case of large speed changes, although this category of methods can efficiently address *cross-view gait recognition* [49][50][52].

In our study, we find that walking speed change is a kind of internal factor caused by a walking person, which will present different efforts on different persons. We cannot treat it in the same way as in the case of view change which is a kind of external factor caused by shooting environments. Consequently, seeking a generic solution in the second category to cover any walking person becomes very difficult. Therefore, this study chooses to seek a solution in the first category.

Figure 1.4: Sample gait images under various views from the CASIA gait database A.

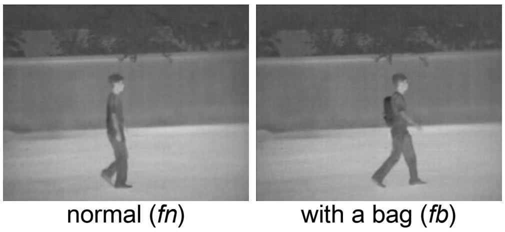


Figure 1.5: Sample gait images under various walking conditions from the CASIA gait database C.

1.3 Gait Databases

This section lists the databases adopted in our study.

1.3.1 The CASIA gait database A

The CASIA gait database A [3] includes 20 subjects from three views, namely frontal, oblique, and lateral views. Each subject walked along a straight-line path back and forth twice. In this way, four videos were recorded for each subject from each view. Figure 1.4 shows sample images from the CASIA gait database A.

1.3.2 The CASIA gait database B

The CASIA gait database B [3] is a large multi-view gait database which contains 124 subjects from 11 views ($0°$, $18°$, ..., $180°$). Under each view, ten gait sequences were captured for each person including six sequences in normal walking (without carrying a bag and without wearing a coat), two sequences in walking when carrying a bag, and two sequences in walking when wearing a coat. Moreover, this

Figure 1.6: Sample gait images under the left and right cameras from the USF gait database.

Figure 1.7: Sample gait images under various walking conditions from the CMU Mobo gait database.

database can directly support the study of gait recognition under view change. Figure 1.2 shows sample images from the CASIA gait database B.

1.3.3 The CASIA gait database C

The CASIA gait database C [56] contains 153 subjects. It was recorded under four walking conditions, namely normal walking (fn), slow walking (fs), fast walking (fq), and normal walking with a bag (fb). Ten gait sequences were recorded for each subject (4 sequences for fn, 2 sequences for fs, 2 sequences for fq, 2 sequences for fb). The videos were all captured at night by infrared (thermal) cameras. Figure 1.5 shows sample images from the CASIA gait database C.

1.3.4 The USF gait database

The USF gait database [2] is considered as a real scene captured in an uncontrolled environment. It is challenging because of several difficulties of the outdoor environment such as wind, shadow, and illumination. This database consists of persons walking in elliptical paths in front of cameras. It contains a set of 12 challenge experiments which are designed to investigate the effect of five factors (i.e. view, surface, shoe, carrying condition, time) affecting the performance of gait recognition. Moreover, 122 subjects were recorded under two views/cameras (L and R). The cameras' lines of sight are verged at approximately 30 degrees. Figure 1.6 shows sample images from the USF gait database.

1.3.5 The OU-ISIR gait database

The OU-ISIR gait database [9] contains 6 different walking speeds from 2 km/h to 7 km/h with 1 km/h interval. Total 31 subjects are used in this study. Two video sequences were recorded for each subject from each speed. Figure 1.3 shows sample images from the OU-ISIR gait database.

1.3.6 The CMU Mobo gait database

The CMU Mobo gait database [57] contains 25 subjects from 2 different walking speeds, namely slow walking (3.3 km/h) and fast walking (4.5 km/h). The videos were captured in an indoor scenario. Each subject was recorded under four walking conditions, slow walking, fast walking, slow walking at a certain slope, and slow walking when holding a ball. Figure 1.7 shows sample images from the CMU Mobo gait database.

1.3.7 Overall comparisons on various gait databases

Table 1.2 shows comparisons of full details between different adopted gait databases.

1.4 Contributions

This thesis mainly addresses gait recognition under changes of walking conditions focusing on an observed view and a walking speed. The other relevant problems are also sorted out, including *gait period* analysis, *view classification*, *walking speed estimation*, optimal view selection, and typical implementations of gait recognition under view change for different cases such as *cross-view gait recognition* and *multi-view gait recognition*.

Table 1.2: Comparisons between different adopted gait databases.

Database	Number of subjects	Number of sequences	Environment	Year	Walking variations
CASIA-A	20	240	outdoor	2001	view
CASIA-B	124	13,640	indoor	2005	view, cloth carrying
CASIA-C	153	1,530	outdoor	2005	speed, carrying
USF	122	1,870	outdoor	2001	view, surface shoe, carrying, time
OU-ISIR	34-185	$> 2{,}000$	indoor	2007	view, speed, cloth, gait fluctuation
CMU	25	600	indoor	2001	view, speed, carrying, surface

In this thesis, the first method is proposed for a robust gait recognition in order to deal with changes of various walking conditions including walking speed, view, carrying condition, floor type, cloth type, etc. *WBP* is developed as a new robust gait feature upon *GEI*. Although *GEI* holds very rich gait information, it contains many uncertainties across different walking conditions. To construct *WBP*, partial ideas of *LBP* are applied to *GEI* to generate a binary pattern. The pattern comprehensively describes features of *GEI* and also significantly reduces such uncertainties. To enhance the recognition performance, an adaptive weighting technique is proposed and applied to discriminate significances of bits in the binary pattern.

The proposed *WBP* improves *GEI* to be robust to various walking variations. However, its performance is not fully satisfactory when view changes or walking speed changes become significant. These problems will be addressed in the later proposed methods. Instead of *WBP*, *GEI* is still adopted as a baseline gait feature in our second and third proposed methods. This is because *GEI* adequately captures both temporal and spatial information of walking manners. It is a core used for bridging gaps between gaits from different views/speeds.

To more efficiently solve gait recognition under view change, the second method is proposed based on *VTM* which is used to learn a mapping relationship between gait features observed across views. It belongs to the third category of methods for gait recognition under view change (see Section 1.2.2.3). Two types of *VTM* including *dVTM* and *rVTM* have been developed in this study.

dVTM was introduced in [52] based on a matrix decomposition concept. This method solves the view transformation problem via the equivalent decomposition problem. A 2-D gait matrix is built up such that each row contains gait features from the same view of different subjects and each column contains gait features from the same subject under different views. Then, *SVD* is applied to decompose the gait matrix into view-independent matrix and subject-independent matrix which is used to transform gait features across views.

To improve the performance of the method in [52], we create *dVTM* based on *GEI* optimized by *Linear Discriminant Analysis (LDA)*. Moreover, the truncated *SVD* is also introduced to avoid oversizing and overfitting of *dVTM*. In this way, our method can be more computationally feasible and better able to achieve higher accuracy.

However, the size of *dVTM* still depends on the size of the training dataset. To obtain a reliable *dVTM*, it requires a large training dataset causing a large size of *dVTM* which leads to an unpopular computational complexity. Moreover, the performance of *dVTM* is also bounded by limitations of linear processes on global features.

Thus, we reformulate *VTM* construction as a regression problem in *rVTM* in order to overcome these limitations of *dVTM*. The regression concept is used to reveal correlated motions of gaits from different views. *rVTM* consists of multiple regression processes which are trained to estimate gait feature under one view using correlated information in gait feature(s) under other view(s). The local correlated features under shared motions are exposed using rough body segmentations and correlation coefficients. Three regression techniques are attempted in the construction of *rVTM*, which include *Multi-Layer Perceptron (MLP)*, *Support Vector Regression (SVR)*, and *Sparse Regression (SR)*. Based on our experiments, *SR* is shown to achieve the best performance due to its sparsity constraints.

The proposed *rVTM*-based method can achieve high and reliable performance when compared with other methods in the literature. However, its regression framework is based on mapping between a group of pixels (i.e. called segment) and a single-pixel. Such mapping can become unstable when the pixels shift due to an occlusion and/or a shadow on gait. Besides, the *VTM*-based methods may suffer from degeneracies and singularities caused by features visible in one view but not in others, especially in the case of large view changes.

Sequentially, the third method is proposed to overcome the limitations of the *VTM*-based methods above. It relaxes the mapping relationship across views by using a segment to segment mapping through a correlated motion co-clustering method. The correlation of gaits from different views is then maximized based on the corresponding local segments rather than on the global gait features. In this way, the correlation maximization of gaits from different views will be further optimized by involving only related gait information in the local segments.

In this proposed method, a bipartite graph is used to model correlation between gaits from two different views, then *bipartite graph multipartitioning* is used to co-cluster gaits across the two views into segments. Consequently, the corresponding segments are the most correlated and contain the most similar gait information. Then, *CCA* is applied to project the corresponding segments from the two views into two *CCA* subspaces where their linear correlation is maximized. Finally, a *linear approximation* model is learned to linearly transform the corresponding segments of gaits from the two *CCA* subspaces into the same *CCA* subspace where the similarity will be carried out.

When compared with other methods in the literature, this proposed method can significantly enhance gait recognition under relatively large view change significantly. However, it cannot be efficiently applied for untrained/unseen views as it also belongs to the third category of methods for gait recognition under view change.

To overcome this limitation of the third category, the fourth method is then proposed from a different perspective in the second category (see Section 1.2.2.2). In this method, a new view-invariant gait feature is developed using the following processes. A new *Gait Texture Image (GTI)* is proposed, which represents original gait information on a sequence of gait silhouettes under a certain view. *Transform Invariant Low-rank Textures (TILT)* is applied on *GTI* in order to recover the optimized domain transformation which represents the geometric projection of the gait information from its original view onto the common canonical view. The obtained domain transformation is then applied to project the corresponding gait silhouettes onto the canonical view. In this way, a sequence of gait silhouettes from any view can be normalized onto the common canonical view. Then, *Procrustes Shape Analysis (PSA)* is applied on a sequence of the view-normalized gait silhouettes to construct a novel view-invariant gait feature.

The last proposed method is also based on statistical shape analysis using *PSA* but for gait recognition under speed change in the first category (see Section 1.2.3.1). In order to adopt *PSA* to address the problem of speed change, we carry out complete analyses of impacts on gait recognition caused by various walking speed changes. Accordingly, a new *Higher-order Shape Configuration (HSC)* is proposed to replace a traditional *Centroid Shape Configuration (CSC)* in order to handle the gait shape change caused by walking speed change. *HSC* describes gait shape by using higher-order derivatives of shape boundary such as tangent, curvature, and aberrancy. Based on our analyses, *HSC* is not significantly affected by speed change. In the meantime, we also introduce a *Differential Composition Model (DCM)* which reflects the different effects caused by walking speed change on different body parts. *DCM* further improves discriminability of gait feature using *HSC*-based *PSA*. This proposed method is shown to achieve the current state-of-the-art performance for gait recognition under speed change.

These proposed methods contain different advantages and disadvantages. A proper method can be selected depending on the purpose of a target surveillance application. Also, the methods can be combined for a more efficient system to cover more practical cases.

Moreover, as mentioned above in this section, some other relevant problems are examined along with the proposed methods of gait recognitions under view/speed changes. For example, it is important to estimate a *gait period* because gait is a periodic action and thus should be analyzed within completed *walking cycle*(s). Our proposed method for *gait period* analysis is based on autocorrelation of a waveform containing an aspect ratio (i.e width/height of silhouette bounding box) along a time series of gait sequence. It can efficiently handle all possible views and consumes significantly less computational time than other existing methods which perform the analysis directly on gait silhouettes.

In practice, an automatic *view classification* is essential for gait recognition under view change. It is required to know a view of each probe gait before an appropriate view-normalization process can be applied. Our *view classification* is performed on a complete *LDA* in *PCA* transformed space, which is efficient and robust to outliers. Likewise, a *walking speed estimation* is essential for gait recognition under speed change. In our study, a walking speed is approximately measured as a period (in a unit of frame number) of one gait cycle. Given a video frame rate, such a period can be calculated in a unit of time e.g. second. Then, an absolute walking speed (e.g. m/s, km/h) can be estimated from a reference database.

When a fewer number of cameras is specified for one place, our method is proposed to optimally select cameras' views such that they can efficiently cover any other view through the proposed *VTM*-based methods. Furthermore, we also develop various scenarios for improving gait recognition under view change. For example, in *cross-view gait recognition* through middle view, gaits observed under two different views are transformed into a middle view between the two views, instead of one view into another. It is more accurate to transform gaits between closer views. This is because gaits from a smaller view-difference share more common/correlated information. Also, *multi-view gait recognition* can achieve superior performance, when individual gaits are observed under multiple views. This is because gaits from a greater number of views provide more visual information for matching. Our methods are proposed to deal with gaits from multiple views/cameras through learning processes, without camera cooperation, camera calibration, and frame synchronization. Thus, the proposed methods can be flexible and suitable for most surveillance applications.

1.5 Thesis Outline

Subsequent chapters of this thesis are organized as follows. Chapter 2 discusses necessary background knowledge. Chapter 3 explains pre-processes for gait analysis and the proposed *Weighted Binary Pattern (WBP)* based on *GEI* for a robust gait recognition. In Chapter 4, the *VTM*-based methods for gait recognition under view change are proposed. Comparisons between *dVTM* and *rVTM* are also discussed in term of problem formulations, *VTM* constructions, and recognition performances. In Chapter 5, gait recognition under view change based on correlated motion co-clustering is proposed. In Chapter 6, *PSA*-based gait recognition is discussed. Moreover, complete analyses of impacts on *PSA*-based gait recognition caused by various walking speed changes are carried out. Based on these analyses, the improved scheme of *PSA* is proposed to address gait recognition under speed change. In Chapter 7, we propose a new view-invariant feature for *cross-view gait recognition* based on *TILT* and *PSA*. Finally, Chapter 8 concludes our thesis and discusses some future works.

The works described in Chapters 3, 4, 5, 6, and 7 have been presented in [49][50][53][58][59][60][61][62][63].

Chapter 2

Fundamental Knowledge

This thesis proposes many methods from several different perspectives to address the problems of various walking variations focusing on view changes and speed changes. Thus, many techniques are adopted and adapted accordingly in the proposed methods. This chapter explains and discusses background knowledge of these techniques.

2.1 Gait Energy Image (GEI)

GEI [27] is adopted in this thesis as a baseline gait feature. It captures several key information of human gait including motion frequency, temporal and spatial changes of human body, and global body shape statistic. It also well reflects the gait rhythm. Since *GEI* is created from a gait sequence, it can avoid synchronization difficulties and prevent noises from individual images.

In a window of complete *walking cycle*(s), *GEI* is constructed on the spatial domain as gait feature [27][64]. Given a set of binary gait bounding boxes ($\{B_t(x,y)\}_{t=1}^{N_g}$, where $B_t(x,y)$ is a pixel at position (x,y) of B_t and N_g is the total number of frames from complete *gait period*(s)), *GEI* is obtained as:

$$G(x,y) = \frac{\sum_{t=1}^{N_g} B_t(x,y)}{N_g} \tag{2.1}$$

where the binary gait bounding box is a bounding box that just covers a gait silhouette extracted from each frame. B_t are normalized to the same height and

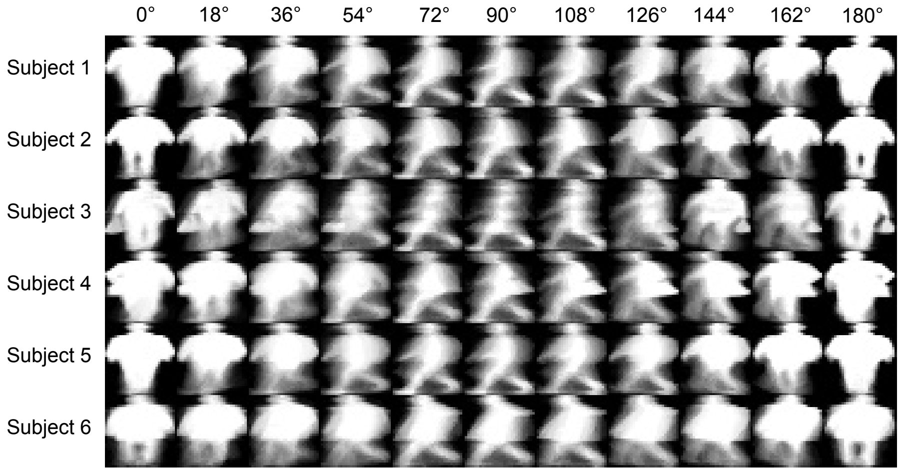


Figure 2.1: Sample *GEI*s under various views, clothes, and carrying conditions. Subjects 1 and 2 are examples of normal walking. Subjects 3 and 4 are examples of walking when carrying a bag. Subjects 5 and 6 are examples of walking when wearing a coat.

their upper half are then aligned along their horizontal centroids, before being used in Equation (2.1).

Figure 2.1 shows examples of *GEI*s from various subjects under various views, clothes, and carrying conditions on the CASIA gait database B. From Figure 2.1, the white color represents a pixel with highest intensity and the black color represents a pixel with lowest intensity. In this way, the lighter gray color represents a pixel with higher intensity. A pixel with higher intensity in *GEI* corresponds to a body part that moves less during a *walking cycle* (e.g. head, torso), while a pixel with lower intensity corresponds to a body part that moves constantly (e.g. legs, arms).

Moreover, it can be seen that *GEI*s vary on different views, clothes, and carrying conditions. Therefore, it is difficult and inefficient to directly measure similarity of *GEI*s across different walking conditions. More efforts are required upon *GEI*s in order to make them more robust to changes of walking conditions, or to normalize them onto a common/shared feature space.

In this thesis, the method proposed in Chapter 3 is a type to enhance a more robust *GEI*. While, the methods proposed in Chapters 4 and 5 are to normalize *GEI*s recorded from various views.

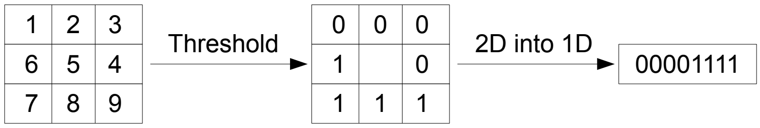


Figure 2.2: The basic *LBP* operator.

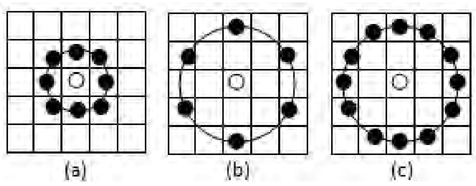


Figure 2.3: Examples of multi-scale *LBP* operator. (a) The circular (8,1) neighborhood. (b) The circular (8,2) neighborhood. (c) The circular (12,2) neighborhood. The pixel values are bi-linearly interpolated when the sampling point is not located at the point with integer coordinate.

2.2 Local Binary Pattern (LBP)

LBP [65][66] is a powerful feature for texture classification. In Chapter 3 of this thesis, it is applied to describe contents of *GEI* which can be considered as a grey texture image. *LBP* can improve the quality of *GEI* as it is capable of a grey-scale invariant texture measurement. This makes *GEI* more robust to walking variations.

LBP is an operator that describes the surrounding of a pixel by generating a bit-code from binary derivatives of the pixel. Figures 2.2 and 2.3 show examples of the *LBP* operator.

In the simplest case, *LBP* takes 3×3 pixels surrounding of a center pixel (see Figure 2.3(a)). Next is to compare each pixel to the center pixel and generate a binary 1 if it has larger value than the center pixel, otherwise the operator will generate a binary 0. This gives an 8-digit binary number. It is usually converted to a decimal for the convenience. This process is applied on every 3×3 blocks of an image to generate a bunch of binary numbers. Then, a histogram is computed to count the frequency of each number. Such histogram is used as a feature vector representing the image. It can describe local structures such as edges, lines, spots, flat areas, and corners.

In the method proposed in Chapter 3, instead of the histogram, the sequence of binary numbers will be directly used to construct gait feature. This is because we still want to preserve spatial information of walking manners which is very important for recognizing gaits.

2.3 Subspace Analysis

PCA and *LDA* are well-known subspace learning techniques. In this thesis, they are used for dimensionality reduction, classification, and feature optimization. Their fundamental concepts are briefly explained as belows.

2.3.1 Principal Component Analysis (PCA)

PCA is a well-known unsupervised technique for dimensionality reduction [67][68]. It will be used in the proposed method for *view classification* in Chapter 4. *PCA* can be done by eigenvalue decomposition of a data covariance (or correlation) matrix or *Singular Value Decomposition (SVD)* of a data matrix [67].

Define a data matrix X with zero empirical mean, where each of its N rows represents a different repetition of $M-$dimensional vector data. The zero empirical mean is resulted from a process where the empirical (sample) mean of the distribution has been subtracted from the dataset.

An eigen-decomposition of the covariance matrix XX^T is performed to produce a matrix $W = (w_1, ..., w_N)$, where w_j are the eigenvectors arranged by their eigenvalues in decreasing order. On the other hand, W can also be computed by applying *SVD* on X such that $X = W\Sigma V^T$ where the matrix Σ is an $M \times N$ rectangular diagonal matrix with non-negative real numbers on the diagonal, and the $N \times N$ matrix V is the matrix of eigenvectors of $X^T X$. The detail of *SVD* is explained in Section 2.4.

For a reduced-dimensionality representation, the W space can be reduced to W_m by selecting only first m ($m << M$) eigenvectors with the m largest eigenvalues. For example, each *GEI* (G_i) can be projected into *PCA* subspace as follow.

$$\acute{G}_i = W_m^T G_i \tag{2.2}$$

2.3.2 Linear Discriminant Analysis (LDA)

Theoretically, *LDA* [68] is a classical statistical approach used for supervised dimensionality reduction, classification, and feature extraction. In Chapter 4 of this thesis, it will be used for two purposes: 1) dimensionality reduction and feature optimization of a gait feature (i.e. *GEI*); 2) *view classification*. Thus, in the proposed *view classification* method, both *PCA* and *LDA* are applied. This is because, unlike *LDA*, *PCA* is not optimized for a class separability [69]. Benefits from both tools are combined for our method.

LDA computes an optimal projection by searching the directions for maximum discrimination of classes. The projection is done by minimizing the trace of within-class scatter matrix (S_w) and maximizing the trace of between-class scatter matrix (S_b) simultaneously. However, since the scatter matrices are proportional to the covariance matrix (S_t), $S_t = S_b + S_w$, *LDA* process is equivalent to minimizing the trace of S_t and maximizing the trace of S_b simultaneously. Therefore, *LDA* considers maximizing of the following objective to obtain the projection matrix W [70].

$$J(W) = \frac{W^T S_b W}{W^T S_w W} \tag{2.3}$$

This is equivalent to solving the following generalized eigensystem [70].

$$S_b w = \lambda S_w w \tag{2.4}$$

Therefore, $W = (w_1, ..., w_N)$ are eigenvectors of $(S_w)^{-1} S_b$. Similarly to *PCA*, to reduce the space dimension, only m ($m << M$) eigenvectors with the m largest eigenvalues are selected.

2.4 Singular Value Decomposition (SVD)

In the linear algebra, *SVD* [71][72] is a factorization of a real or complex matrix, with many useful applications in signal processing and statistics. In this thesis, *SVD* is applied to factorize the gait matrix regarding view factor. This process is used for the *dVTM* construction in Chapter 4.

Formally, SVD factorizes an $M \times N$ matrix X into the following form.

$$X = U\Sigma V^T \tag{2.5}$$

where U is an $M \times M$ unitary matrix, Σ is an $M \times N$ rectangular diagonal matrix with non-negative real numbers on the diagonal, and V^T (i.e. the conjugate transpose of V) is an $N \times N$ unitary matrix. The matrix Σ contains the singular values of X. The M columns of $U = \{u_i\}_{i=1}^{M}$ and the N columns of $V = \{v_i\}_{i=1}^{N}$ are called the left-singular vectors and right-singular vectors of X, respectively.

Moreover, SVD can be solved via the eigen-decomposition. The singular values of X (i.e. the diagonal entries of Σ) are the square roots of the eigenvalues of both X^TX and XX^T. The left-singular vectors (u_i) and right-singular vectors (v_i) of X can be computed by solving the following generalized eigensystems.

$$\begin{aligned} XX^T u_i &= (\Sigma_{i,i})^2 u_i \\ X^T X v_i &= (\Sigma_{i,i})^2 v_i \end{aligned} \tag{2.6}$$

where $\Sigma_{i,i}$ is a diagonal entry (i, i) of Σ. Thus, the left-singular vectors u_i are eigenvectors of XX^T and the right-singular vectors v_i are eigenvectors of X^TX.

To construct $dVTM$ in Chapter 4, the matrix X is built up in such a way that each row contains gait features from same view but N different subjects, and each column contains gait features from same subject but M different views. In this way, the matrix U will be independent to subject but dependent to view, and the matrix V will be independent to view but dependent to subject. Thus, the matrix U will be used further for view transformation. It will also be generic to any subject.

2.5 Regression Tools

To construct $rVTM$ in Chapter 4, a regression is used to model correlated motions of gaits from different views. Three regression techniques are attempted in this study. Given a training dataset as $\{D_i : (X_i, y_i) \mid X_i \in \mathbb{X},\ y_i \in \Re,\ i = 1...N\ \}$, where N is the number of training samples, y is the output (i.e. response) in real value, $\mathbb{X}$ denotes the space of input patterns, and $X = \{x_p\}_{p=1}^{P}$ where P is the dimension of the input space (i.e. P predictors). The regression equation $f(X)$ for describing the case of linear function is defined as follow.

$$y = f(X) = < W, X > + b \tag{2.7}$$

where W is a weight vector such that $W = \{w_p\}_{p=1}^{P}$, $b \in \Re$, and $< \cdot, \cdot >$ denotes the dot product. The concepts of the three regression techniques (i.e. *MLP*, *SVR*, *SR*) are briefly explained to describe the standard regression equation in (2.7) as belows.

2.5.1 Multi-Layer Perceptron (MLP)

MLP [73][74][75][76] is selected as a regression tool for the $rVTM$ construction because of the following advantages. First, *MLP* has an ability to be used as an arbitrary function approximation mechanism which learns from the training data. Therefore, with sufficient training gait data, *MLP* can be used to non-linearly create reliable $rVTM$s. Second, such $rVTM$s generated from *MLP* can be extremely robust if the model structure, cost function, and learning algorithm are appropriately selected. Third, *MLP* can be used naturally in online learning and large dataset applications due to its fast and parallel implementation on hardware.

The typical perceptron describes Equation (2.7) as follow.

$$\begin{aligned} y &= \phi < W,\ X > + b \\ &= \phi\ (\sum_{p=1}^{P} w_p\ x_p) + b \end{aligned} \tag{2.8}$$

where ϕ is a transfer function for artificial neural network. It makes the network to perform a specific task regarding how the units are connected to one another. *MLP* is a modification of the typical perceptron such that it uses three or more layers of neurons with non-linear activation functions. For example (see Figure 2.4), *MLP* with three main layers including input layer, one hidden layer, and output layer is described in Equation (2.9).

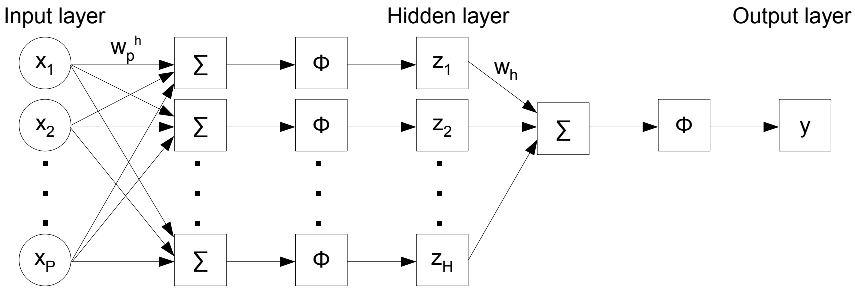


Figure 2.4: Example of the typical neural network with one hidden layer and one output node.

$$
\begin{aligned}
y &= \phi < W,\ (\phi < W^h,\ X > + b_h) > + b \\
&= \phi\ (\sum_{h=1}^{H} w_h\ (\phi\ (\sum_{p=1}^{P} w_p^h\ x_p) + b_h)) + b
\end{aligned}
\tag{2.9}
$$

where H is the number of neurons in hidden layer, w_h and b_h are a weight vector and a bias for the h^{th} neuron in hidden layer respectively, and $W^h = \{w_p^h\}_{p=1}^P$ is a weight vector in input layer that corresponds to the h^{th} neuron in hidden layer. MLP can contain more than one hidden layer by logically extend Equation (2.9).

MLP performs a biased weighted sum of inputs (X) and passes this activation level through transfer function (ϕ) to produce the corresponding output (y). A regular choice of the transfer function is a sigmoid function as $\phi(d) = 1/(1 + exp(-d))$. Although the network has a simple interpretation as a form of input-output model, it can model functions of almost arbitrary complexity.

In the training phase, the network's weights and thresholds must be adjusted to minimize the prediction errors. A well recognized example of a neural network training algorithm is back-propagation by Gradient Descent [73]. A sufficient number of training input-output pairs are needed to estimate the regression's parameters in order to obtain the regression model.

2.5.2 Support Vector Regression (SVR)

As a regression tool, SVR [77][78][79][80] has several advantages. First, it features good generalization performance. It is a core requirement for most regression

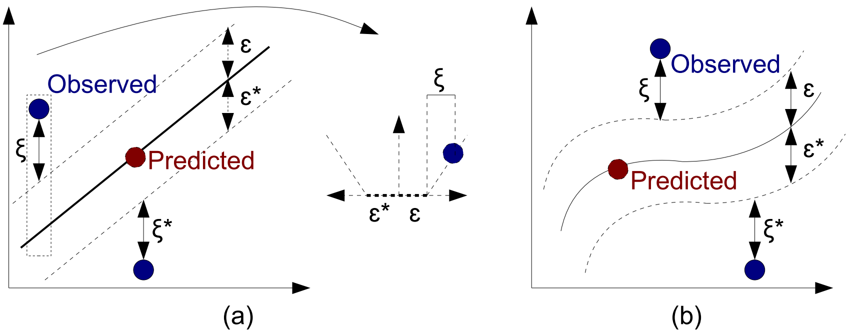


Figure 2.5: The soft margin lose settings for (a) a linear SVM and (b) a nonlinear SVM.

applications including this study. Second, its representation is sparse. This is because a regression model obtained by SVR depends only on support vectors, a subspace of the training dataset. Therefore, the dimension of $rVTM$ is sparse and controllable. Third, unlike other regression tools, SVR does not have a local minimum problem. Its solution is unique and globally optimal. Hence, we can obtain a global optimized $rVTM$ based on the supplied context of the training data. Fourth, SVR is a kernel-based regression technique. Thus, a non-linear kernel can transform an input space to an arbitrary high-dimension feature space, which provides a better discriminality for the regression.

As shown in Figure 2.5, ϵ-SVR attempts to find a function $f(X)$ that satisfies the following three aspects [77]. First, the error from the difference between predicted and observed values is disregarded on the condition that it is less than ϵ. Second, the soft margin loss function is allowed. Slack variables ξ, ξ^* are introduced to cope with predicted values that lie outside the absolute ϵ region. Third, the flatness of W is attempted by minimizing the norm $||W||^2 = <W, W>$. The three aspects can be written as a convex optimization problem as in Equation (2.10).

$$
\begin{aligned}
&minimize \quad \frac{1}{2}||W||^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\
&subject\ to \quad \begin{cases} y_i - <W, X_i> - b & \leq \quad \epsilon + \xi_i \\ <W, X_i> + b - \ y_i & \leq \quad \epsilon + \xi_i^* \\ \xi_i,\ \xi_i^* & \geq \quad 0 \end{cases}
\end{aligned}
\tag{2.10}
$$

where the constant C determines the trade-off between the flatness of f and the amount up to which deviations larger than ϵ are tolerated.

The optimization problem in Equation (2.10) can be solved more easily in its dual formulation. Thus, a standard dualization method utilizing Lagrange multipliers is used and proceeded as follow.

$$\begin{aligned} L =& \frac{1}{2}||W||^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) - \sum_{i=1}^{N}(\eta_i\xi_i + \eta_i^*\xi_i^*) \\ & - \sum_{i=1}^{N}\alpha_i(\epsilon + \xi_i - y_i + < W, X_i > + b) \\ & - \sum_{i=1}^{N}\alpha_i^*(\epsilon + \xi_i^* + y_i - < W, X_i > - b) \end{aligned} \tag{2.11}$$

where L is the Lagrangian and η_i, η_i^*, α_i, $\alpha_i^* \geq 0$ are Lagrange multipliers.

It follows from the saddle point condition that the partial derivatives of L with respect to the four primal variables (W, b, ξ_i, ξ_i^*) have to vanish for optimality. Thus, substituting these partial derivatives into Equation (2.11) yield the dual optimization problem as follow.

$$\begin{aligned} & \mathit{maximize} \quad \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) < X_i, X_j > \\ -\epsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y_i(\alpha_i - \alpha_i^*) \end{cases} \\ & \mathit{subject\ to} \quad \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0,\ \ \alpha_i, \alpha_i^* \in [0, C] \end{aligned} \tag{2.12}$$

Support vector expansion can be obtained from the above process such that W can be completely described as a linear combination of the training pattern X as follow.

$$W = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)X_i \tag{2.13}$$

Substitute (2.13) into (2.7), then the complete algorithm can be described in terms of dot products between the data as follow.

$$f(X) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) < X_i, X > + b \tag{2.14}$$

The complexity of a function's representation by SVR is independent of the dimensionality of the input space $\mathbb{X}$, and depends only on the number of support vectors.

The SVR described above is a linear kernel-based SVR. To apply a non-linear kernel to SVR, in Equations (2.12) and (2.14), the dot products $< X_i, X_j >$ for linear kernel are replaced with kernel $k(X_i, X_j)$. Two non-linear kernels [81] are used in this study. The first kernel is a polynomial kernel of degree d which can be defined as follow.

$$K_{d,s,k}^{polynomial}(X_i, X_j) = (s < X_i, X_j > + k)^d \tag{2.15}$$

where d is the degree of polynomial, s is the scale coefficient, and k is the bias.

The degree of the polynomial kernel controls the flexibility of the resulting classifiers. When $d = 0$, it is in fact a linear kernel, which is not sufficient when a nonlinear relationship between features exist. The second kernel that is widely used is the Gaussian kernel or Radius Basic Function (RBF) kernel. The RBF is defined as follow.

$$K_{\sigma}^{RBF}(X_i, X_j) = exp(-\frac{1}{\sigma}||X_i - X_j||^2) \tag{2.16}$$

The $\sigma > 0$ is a parameter that controls the width of the Gaussian, which plays a similar role to the degree of the polynomial kernel. σ in the Gaussian kernel and d in the polynomial kernel determine the flexibility of the produced SVR in fitting the data. A larger d or smaller σ may lead to overfitting.

2.5.3 Sparse Regression (SR)

To further improve the motion feature selection and the quality of regression training processes for the $rVTM$ construction, *SR* [82][83] by the elastic net is adopted in place of *MLP* and *SVR*. The embedded feature selection in *SR* can be used to refine the motion feature selection in order to obtain more stable fitted models in *rVTM*. This section provides fundamental knowledge of *SR*.

SR can be achieved using the lasso [82][84][85]. The lasso is a penalized least squares method subject to a bound on L_1 norm of regression coefficients. Thus, the lasso estimates $\hat{W}_{lasso}$ by minimizing the lasso criterion as follow.

$$\hat{W}_{lasso} = \underset{W}{\operatorname{argmin}} ||y - \sum_{p=1}^{P} x_p w_p||_2^2 + \lambda \sum_{p=1}^{P} ||w_p||_1 \tag{2.17}$$

where λ is a non-negative value, $||\cdot||_1$ is L_1 norm, and $||\cdot||_2^2$ is the square of L_2 norm ($||\cdot||_2$). The lasso provides both continuous shrinkage and automatic variable selection at the same time. Due to nature of L_1 penalty, some coefficients (w_p) can be shrunk to exact zero if λ is large enough.

However, the lasso has several limitations [83]. The one which will affects this study is that the lasso tends to arbitrarily select only one predictor from a group of predictors that contains very high pairwise correlations. The resulting regression model will become difficult because it may lose many useful predictors which still are associated with the response. This will cause an unreliable *rVTM* construction.

The elastic net [83] generalizes the lasso to overcome its drawbacks. The elastic net penalty is a convex combination of ridge penalty (L_2 norm) and lasso penalty (L_1 norm). Thus, the elastic net estimates $\hat{W}_{elastic}$ as:

$$\begin{aligned}\hat{W}_{elastic} = (1+\lambda_2) \underset{W}{\operatorname{argmin}} ||y - \sum_{p=1}^{P} x_p w_p||_2^2 \\ + \lambda_2 \sum_{p=1}^{P} ||w_p||_2^2 + \lambda_1 \sum_{p=1}^{P} ||w_p||_1\end{aligned} \tag{2.18}$$

where λ_1 and λ_2 are non-negative values. The elastic net can potentially encourage a grouping effect, where strong correlated predictors tend to be kept in the model.

The ridge penalty is used to shrink coefficients of correlated predictors towards each other. While the lasso penalty is somewhat indifferent to highly correlated predictors and will tend to pick one and ignore the rest by shrink their coefficients to zero. Through the combination of two penalties in the elastic net, they can select sufficient predictors with satisfying low redundancy to generate stable and non-overfitting regression models for the *rVTM* construction in our proposed method.

2.6 Bipartite Graph Partitioning

A bipartite graph partitioning [86][87] is adopted in Chapter 5 to co-cluster correlated segments of gait features across different views. Then, the view-normalization process will be applied on these local gait segments instead of global gait features. This will enhance the performance of the view-normalization process for gait recognition under view change.

A bipartite graph is defined as $B = (V, E, W)$ where $V = S_1 \cup S_2$ contains two sets of vertices, $E = \{(a, b) : a \in S_1, b \in S_2\}$ is the set of edges, and $W = \{w_{a,b}\}$ is the sets of edge-weights. The classical *bipartite graph bipartitioning* is to partition the graph vertices (V) into 2 clusters as $V_1 = S_1^1 \cup S_2^1$ and $V_2 = S_1^2 \cup S_2^2$ where $S_1 = S_1^1 \cup S_1^2$ and $S_2 = S_2^1 \cup S_2^2$, by minimizing the following objective function.

$$cut(V1, V2) = f(S_1^1, S_2^2) + f(S_1^2, S_2^1) \tag{2.19}$$

where $f(S_1^1, S_2^2) = \sum_{a \in S_1^1, b \in S_2^2} w_{a,b}$ and $f(S_1^2, S_2^1) = \sum_{a \in S_1^2, b \in S_2^1} w_{a,b}$. This objective function can be approximately solved by spectral clustering, which starts from constructing the Laplacian matrix L ($|V| \times |V|$ symmetric matrix) as in Equation (2.20).

$$L(i,j) = \begin{cases} \sum_{m=1}^{|V|} w_{V(i),V(m)} & , i = j \\ -w_{V(i),V(j)} & , i \neq j \textit{ and there is an edge } (i,j) \\ 0 & , \textit{otherwise} \end{cases} \tag{2.20}$$

where $V(i)$ is the i^{th} vertex in the set of vertices V.

The graph bipartitioning can be understood as the projection of vertices onto two points +1 or -1 in the discrete case. A good projection shall minimize $\sum_{(a,b)\in E} w_{a,b}(q_a - q_b)^2$ where q_a is the projection value of vertex a such that

$q_a = +1$ if $a \in V_1$ or $q_a = -1$ if $a \in V_2$. Intuitively, larger $w_{a,b}$ will result in higher possibility of projecting vertices a and b onto a same point. Thus, given L and a partition vector $\mathbf{q} = \{q_a\}_{a \in V}$, the objective function for graph bipartitioning can be obtained as the following equation.

$$cut(V_1, V_2) = \frac{1}{4}\mathbf{q}^T L\mathbf{q} = \frac{1}{4}\sum_{(a,b)\in E} w_{a,b}(q_a - q_b)^2 \tag{2.21}$$

However, the objective function should not only minimize cut value but also balance a partition. The normalized graph cut ($nCut$) is introduced as follow.

$$nCut(V_1, V_2) = \frac{cut(V_1, V_2)}{f(S_1^1, S_2^1)} + \frac{cut(V_1, V_2)}{f(S_1^2, S_2^2)} \tag{2.22}$$

The partition that minimizes $nCut(V_1, V_2)$ is the optimal partition of vertices ($V = V_1 \cup V_2 = (S_1^1 \cup S_2^1) \cup (S_1^2 \cup S_2^2)$) in graph B. It has been proved that the second smallest eigenvector of the generalized eigenvalue problem in Equation (2.23) provides a real relaxation to the discrete optimization problem of finding the minimum normalized cut.

$$Lz = \lambda Dz \tag{2.23}$$

where

$$L = \begin{bmatrix} D_1 & -A \\ -A^T & D_2 \end{bmatrix}, D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \tag{2.24}$$

where A is the $|S_1| \times |S_2|$ matrix such that $A(i,j) = w_{S_1(i),S_2(j)}$, $S_1(i)$ and $S_2(j)$ are the i^{th} vertex and j^{th} vertex in the sets of vertices S_1 and S_2 respectively, D_1 and D_2 are the diagonal matrices such that $D_1(i,i) = \sum_{j=1}^{|S_2|} A(i,j)$ and $D_2(j,j) = \sum_{i=1}^{|S_1|} A(i,j)$.

2.6.1 Bipartite graph bipartitioning via SVD

The eigenvalue problem in Equation (2.23) can be solved more computationally efficient via *SVD*. To do this, Equation (2.23) is first rewritten (by substitution of L and D from Equation (2.24) and $z = [x, y]^T$) as follow.

$$\begin{bmatrix} D_1 & -A \\ -A^T & D_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.25}$$

Assume that D_1 and D_2 are nonsingular, $u = D_1^{1/2}x$ and $v = D_2^{1/2}y$, Equation (2.25) is then rewritten as follow.

$$\begin{aligned} D_1^{-1/2} A D_2^{-1/2} v = (1-\lambda)u, \\ D_2^{-1/2} A^T D_1^{-1/2} u = (1-\lambda)v, \end{aligned} \tag{2.26}$$

Equation (2.26) is clearly defined as *SVD* of the normalized matrix $A_n = D_1^{-1/2} A D_2^{-1/2}$ where u and v are the left and right singular vectors respectively, and $(1-\lambda)$ is the corresponding singular value. Thus, instead of computing the eigenvector corresponding to the second smallest eigenvalue of Equation (2.23), we can equivalently compute the left and right singular vectors (u_2 and v_2) corresponding to the second largest singular value of A_n to provide a real relaxation to *bipartite graph bipartitioning*. Computationally, working on A_n is much faster since A_n is of size $|S_1| \times |S_2|$ while the matrix L is of size $2|S_1| \times 2|S_2|$.

Based on Equation (2.26) and the construction of A, each 1-D vector in u corresponds to a vertex in S_1 and each 1-D vector in v corresponds to a vertex in S_2. Any proper partitioning method (e.g. k-mean adopted in this study) can be applied to split 1-D vectors in $[u, v]^T$ into two groups. Accordingly, the corresponding vertices in S_1 and S_2 will be partitioned into two clusters as mentioned above.

2.6.2 Bipartite graph multipartitioning via SVD

The above bipartitioning method can be extended for solving more general problem of the multipartitioning. Just as the second left and right singular vectors (u_2 and v_2) contain bi-modal information for bipartitioning, l left and right singular vectors ($\mathbf{U}$ and $\mathbf{V}$) often contain N_c-modal information for N_c-partitioning (i.e. multipartitioning) where $l = \lceil log_2\, N_c \rceil$. As defined herein, $\mathbf{U} = [u_2, ..., u_{l+1}] \in \Re^{|S_1| \times l}$ and $\mathbf{V} = [v_2, ..., v_{l+1}] \in \Re^{|S_2| \times l}$ where u_i and v_i are the left and right singular vectors respectively corresponding to the i^{th} largest singular value of A_n.

At this stage, through the singular vectors in $\mathbf{U}$ and $\mathbf{V}$ as discussed above, the eigenvectors (Z) which relax the bipartite graph partitioning can be obtained as follow.

$$Z = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} D_1^{-1/2}\mathbf{U} \\ D_2^{-1/2}\mathbf{V} \end{bmatrix} \tag{2.27}$$

Each l-dimensional vector in Z is a representative information of each vertex in S_1 and S_2 of the graph B. According to Equation (2.27) and the constructions of U and V, the first $|S_1|$ vectors in Z correspond to vertices in S_1 and the rest of $|S_2|$ vectors in Z correspond to vertices in S_2.

Then, a simple clustering algorithm such as k-mean ($k = N_c$) can be applied on these l-dimensional vectors in Z to cluster them into N_c groups which sequentially provide the multipartitioning of the corresponding vertices (V) of the graph B as $\{S_1^k \cup S_2^k\}_{k=1}^{N_c}$.

2.7 Canonical Correlation Analysis (CCA)

CCA [88][89] is used in Chapter 5 to maximize linear correlations of correlated segments of gait features across different views. This process is applied to normalize gait features onto a common *CCA* subspace where gait similarity can be measured accordingly.

CCA seeks a pair of direction w_X and w_Y such that the correlation ρ between the two projections ${w_X}^T X$ and ${w_Y}^T Y$ is maximized, where X and Y are two sets of variables.

$$\rho = \frac{E[{w_X}^T X Y^T w_Y]}{\sqrt{E[{w_X}^T X X^T w_X]}\sqrt{E[{w_Y}^T Y Y^T w_Y]}} \tag{2.28}$$

where E stands for the expected value. Equation (2.28) can be rewritten in terms of covariance matrices as follow.

$$\rho = \frac{w_X^T C_{XY}\ w_Y}{\sqrt{w_X^T C_{XX}\ w_X w_Y^T C_{YY}\ w_Y}} \tag{2.29}$$

where C_{XX} and C_{YY} are the within-set covariance matrices of X and Y respectively and C_{XY} is the between-set covariance matrix of X and Y. *CCA* maximizes ρ by solving the derivative of Equation (2.29) and setting it to zero. This yields the following eigenvalue equations.

$$\begin{aligned} C_{XX}^{-1}\ C_{XY}\ C_{YY}^{-1}\ C_{YX}\ w_X = \rho^2 w_X \\ C_{YY}^{-1}\ C_{YX}\ C_{XX}^{-1}\ C_{XY}\ w_Y = \rho^2 w_Y \end{aligned} \tag{2.30}$$

where the square canonical correlation ρ^2 is an eigenvalue, and $w_X \in \Re^{|X|\times 1}$ and $w_Y \in \Re^{|Y|\times 1}$ are eigenvectors.

2.8 Procrustes Shape Analysis (PSA)

In this thesis, *PSA* [90][91][92] is employed in Chapters 6 and 7 for gait recognition (i.e. gait feature extraction using Procrustes Distance (PD) and gait similarity measurement using Procrustes Mean Shape (PMS)). According to our research, *PSA* is helpful for analyzing gait shapes, which may be inconsistent due to several factors such as: 1) various poses and sizes throughout a *walking cycle* and/or a camera's viewpoint; 2) inconsistency of individual walking patterns; 3) change of gait appearances caused by change of observed view; and 4) change of walking patterns caused by change of walking speed. In this study, an improved scheme of *PSA* will be proposed to tolerate these inconsistencies.

PSA is a process of performing shape preserving Euclidean transformation on a set of shapes. It is able to achieve similarity measurement between two sets of shapes by properly superimposing. This property is useful for gait recognition. During the superimposition, the positions and the sizes of gait shapes are adjusted by proper translation, rotation, and scaling.

The core idea of *PSA* is to find the best way to superimpose one shape onto another shape, by minimizing the Euclidean distance of their shape configurations (Z_1 and Z_2) as follow.

$$\min_{\alpha,\ \beta} \|Z_1 - \alpha 1_k - \beta Z_2\|^2,\ \ \beta = |\beta| e^{j\angle\beta} \tag{2.31}$$

where $\alpha 1_k$ represents translation, $|\beta|$ and $\angle\beta$ represent scaling and rotation of Z_2 respectively.

To simplify the discussion, we consider a regular case where two shapes in Equation (2.31) are invariant to translation. For example, they are registered at their shape centroids. Then, the solution of Equation (2.31) can be obtained as follow.

$$\alpha = 0, \;\; \beta = \frac{|Z_1^* Z_2|^2}{||Z_2||^2} \tag{2.32}$$

where the superscript $*$ represents a complex conjugation transpose. The β presents the similarity between two shape configurations. Therefore, in the framework of *PSA*, Procrustes Distance (PD) $d_P(Z_1, Z_2)$ is used to quantify the dissimilarity of two shape configurations as follow.

$$\begin{aligned} d_P(Z_1, Z_2) &= 1 - \operatorname*{argmin}_{\beta} \|Z_1 - \beta Z_2\|^2 \\ &= 1 - \frac{|Z_1^* Z_2|^2}{||Z_1||^2 ||Z_2||^2} \end{aligned} \tag{2.33}$$

where the β has been normalized as a value between 0 and 1. Therefore, the dissimilarity is simply the subtraction of the similarity from one.

Moreover, *PSA* provides a way to define Procrustes Mean Shape (PMS) as an average shape configuration (Z) for a given set of N shape configurations $\{Z_i\}_{i=1}^{N}$. PMS is estimated by minimizing a sum of PDs between the mean shape (Z) and each configuration (Z_i) in the set as follow.

$$\min_{\alpha_i,\ \beta_i} \sum_{i=1}^{N} ||Z - \alpha_i 1_k - \beta_i Z_i||^2, \qquad \beta_i = |\beta_i| e^{j\angle\beta_i} \tag{2.34}$$

where $\alpha_i 1_k$ gives the translation of Z_i, and $|\beta_i|$ and $\angle\beta_i$ gives the scale and the rotation of Z_i, respectively. Again, the PMS analysis can be further simplified by assuming that the shape configurations are translation-invariant. Then, the mean shape (Z) can be calculated without iteration as the dominant eigenvector of the complex sum of squares and products matrix S_Z. The proof can be found in [91].

$$S_Z = \sum_{i=1}^{N} (Z_i Z_i^*)/(Z_i^* Z_i) \tag{2.35}$$

2.9 Transform Invariant Low-rank Textures

Transform Invariant Low-rank Textures (TILT) [93] aims to extract invariant information from regions in a 2D image that correspond to a very rich class of regular patterns on a planar surface in 3D, whose appearance can be modeled approximately as a low-rank matrix. Such low-rank textures capture geometrically meaningful structures in an image. They includes conventional local features such as edges, corners, symmetric patterns, and all kinds of regular.

These low-rank textures also exist in gait analysis. For example, edges and corners can be found in an gait image (e.g. *Gait Energy Image (GEI)* [27], Gait Flow Image (GFI) [48], frequency-domain gait feature using Fourier Transfer (FT) [52]) representing a video gait sequence of complete *walking cycle*(s). Particularly, regular and symmetric patterns can be also found in gait since it is a periodic action which contains repeated behavior.

In fact, an image of such a texture may be deformed by the camera projection and undergoes certain domain transformation (e.g. affine or projective). The transformed texture is no longer low-rank in the image domain. Thus, *TILT* is to simultaneously recover such a low-rank texture from its deformed image, and the associated deformation in a form of domain transformation. This is performed using advanced convex optimization tools from the matrix rank minimization.

In gait analysis, the gait image can be deformed by change of observed view. In Chapter 7 of this thesis, *TILT* will be used to recover domain transformation which represents 3D geometry of a planar region in the gait image recorded under a certain view. In this way, gait from any view can be normalized onto the common canonical view (i.e. approximately side view) where low-rank texture of the gait image is obtained, by using its corresponding recovered domain transformation. The view-normalized gait will be used to generate a novel view-invariant gait feature.

In this study, a 2D texture is considered as a function $I^0(x, y) \in \Re^2$. I^0 is a low-rank texture if the family of one-dimensional functions $\{I^0(x, y_0)|y_0 \in \Re\}$ span a finite low-dimensional linear subspace as follow.

$$r = dim(span\{I^0(x, y_0)|y_0 \in \Re\}) \leq k \tag{2.36}$$

where k is a small positive integer. If r is finite, then I^0 is referred as a rank-r texture.

Although many surfaces or structures in 3D exhibit low-rank textures, their images do not. Suppose that a low-rank texture $I^0(x, y)$ lies on a planar surface in the scene. The image $I(x, y)$ which is observed from a certain view is a transformed version of the original low-rank texture function $I^0(x, y)$.

$$I(x, y) = I^0 \circ \tau^{-1}(x, y) = I^0(\tau^{-1}(x, y)) \tag{2.37}$$

where domain transformation $\tau : \Re^2 \to \Re^2$ belongs to a certain Lie group such as the rotation group $SO(2)$, the 2D affine group $Aff(2)$, and the homography group $GL(3)$.

In addition to domain transformations, the observed texture image can be corrupted by noises and occlusions, or contains some pixels from the surrounding background. Such deviations are modelled as follow.

$$I \circ \tau = I^0 + E \tag{2.38}$$

where E is a sparse error matrix by assuming that only a small fraction of image pixels are corrupted by large errors.

Given a corrupted I observed from a certain view, next is to recover the low-rank texture I^0, the sparse error matrix E, and the domain transformation τ. This naturally leads to the optimization problem as follow.

$$\min_{I^0, E, \tau} rank(I^0) + \gamma ||E||_0 \qquad s.t. \;\; I \circ \tau = I^0 + E \tag{2.39}$$

where $||E||_0$ denotes the number of non-zero entries in E. That is, Equation (2.39) aims to find I^0 with the lowest possible rank and E with the fewest possible non-zero entries that agree with the observed I up to the domain transformation τ. The $\gamma \geq 0$ is a weighting parameter that trades off the rank of the texture versus the sparsity of the error.

The optimization problem in (2.39) is not directly tractable due to the non-convexity of the matrix rank and the NP-hard of ℓ^0-norm. Thus, it is replaced with its convex relaxation. The $rank(\cdot)$ is replaced with the *nuclear norm* and the ℓ^0-norm is replaced with the ℓ^1-norm. This yields a new optimization problem as follow.

$$\min_{I^0,E,\tau} ||I^0||_* + \lambda||E||_1 \qquad s.t. \;\; I \circ \tau = I^0 + E \tag{2.40}$$

where $||\cdot||_*$ denotes the *nuclear norm* or the sum of the singular values, and $||\cdot||_1$ denotes the ℓ^1-norm or the sum of the absolute values of the entires. However, there is still difficulty in (2.40) which is the non-linearity of the constraint $I \circ \tau = I^0 + E$. The practical solution via successive convex programming to the optimization problem in (2.40) is given in [93].

Theoretical considerations in [94] suggest that the weighting parameter λ should be of the form $C/\sqrt{m}$ where C is a constant (typically set to unity) and m is the number of pixels in the image.

Chapter 3

Automatic Gait Recognition using Weighted Binary Pattern on Video

3.1 Introduction

Human identification by recognizing spontaneous gait recorded in real-world setting is a tough and not yet fully resolved problem in biometrics research. In practice, it may be required to recognize gait under various environments that contain important factors affecting a person's normal walking pattern. For example, these factors may include various walking speeds and views, carrying an object when walking, incline of a floor, and uncertainty of the environment itself. Thus, it is crucial to develop a robust gait feature which can be recognized across various walking conditions.

In this chapter, a robust gait recognition method is proposed to tackle these problems. The proposed method is developed upon well-known *Gait Energy Image (GEI)* [27]. *GEI* is a popular gait feature which has been proved to be efficient for recognizing gait. It has been adopted by many methods and cited by more than 350 papers. In this thesis, *GEI* will be used as a key baseline gait feature throughout.

However, *GEI* is not robust to changes of gait shapes caused by changes of walking conditions and/or environments. In order to achieve a more robust gait recognition, a novel method based on *Weighted Binary Pattern (WBP)* is proposed. *WBP* adopts partial ideas of *Local Binary Pattern (LBP)*, but further adjusts them to construct a binary pattern on *GEI* from a sequence of aligned silhouettes. Then, an adaptive weighting technique is applied to discriminate significances of bits in the binary pattern. Finally, a *WBP* operator is used to create the gait feature.

Compared to most of the existing methods in the literature, this method can better deal with gait frequency, local spatial-temporal human pose features, and global body shape statistics. Based on our experiments on several well-known benchmark databases, it is demonstrated that the proposed method achieves high accuracy, but with low complexity and computational time.

The chapter is organized as follows. Section 3.2 explains necessary pre-processes for gait analysis. *WBP* is proposed in Section 3.3. Experimental results are presented in Section 3.4 and the chapter concludes in Section 3.5.

3.2 Pre-processing and Gait Period Analysis

Given a gait sequence from a video, a silhouette can be extracted from each frame [2] using a non-parametric background subtraction technique [95] that is quite robust to lighting changes, camera jitter, and shadows. However, some extracted silhouettes are incomplete due to (minor) segmentation errors. In this study, mathematical morphological operations [96] are used for holes remedy and noise elimination.

Moreover, gait is a kind of periodic action. Thus, it should be analyzed within completed *walking cycle*(s) [50][97]. In our study, the similar method as that one in [97] is applied to determine *gait period* of each gait sequence. This method can efficiently handle the hard case of frontal view. It also performs the analysis on the aspect ratio instead of the whole gait silhouette [98][99][100][101], which will cost significantly less computational time.

A key process of our gait period analysis is to create a waveform of aspect ratio: width/height of silhouette bounding box along the time series of gait sequence. It is often difficult to directly estimate *gait period* from the waveform, especially for the case of frontal view. Thus, the normalization and autocorrelation are applied to the waveform to obtain clearer repeated curve pattern which can reliably indicate *gait period* (see Figure 3.1). The autocorrelation process is briefed as below.

The *gait period* (T_{frame}) can be detected by maximizing the autocorrelation of the normalized waveform of the aspect ratio ($\{a(i)\}_{i=0}^{N_w}$ where N_w is the size of the waveform or the total number of frames in the gait sequence) as:

$$T_{frame} = 2 \times \underset{t}{\operatorname{argmax}} \frac{\sum_{i=0}^{N_t} a(i)a(i+t)}{\sqrt{\sum_{i=0}^{N_t} a(i)^2}\sqrt{\sum_{i=0}^{N_t} a(i+t)^2}} \tag{3.1}$$

where $N_t = N_w - t - 1$ and *gait period* is estimated in a unit of frame number. The autocorrelation signal contains two local peaks in one *gait period* because of the bilateral symmetry of human gait under any view except frontal view [97]. Therefore, the *gait period* is estimated as twice the local peak as shown in Equation (3.1).

The detailed process of the *gait period* analysis is shown in Figure 3.1 using the CASIA gait database A as the case study. Figure 3.1(a) shows three samples of silhouette sequences obtained under side view, oblique view, and frontal view respectively. Along the time of image sequences, the aspect ratio of each silhouette bounding box is recorded (Figure 3.1(b)). In order to more precisely estimate the *gait period*, each curve in Figure 3.1(b) is subtracted by its mean value and normalized by its standard deviation. Then, the results are processed by mean filtering [102] (Figure 3.1(c)).

Next is to determine the autocorrelation based on the results in Figure 3.1(c). The autocorrelation between the curve in Figure 3.1(c) and its updated version by shifting itself in some degrees (t) is calculated (see Equation (3.1)). In this study, the offset range t is $-40 \leq t \leq 40$ as shown in Figure 3.1(d). From Figure 3.1(b) and (c), it can be seen that the change of bounding box aspect ration is a kind of periodic signal. Thus, when calculating autocorrelation in Figure 3.1(d), there will be a peak when the offset t equals to a *gait period* or its integer multiple.

To more clearly identify the period transition position, first order derivative (Figure 3.1(e)) is calculated based on the results of Figure 3.1(d). The period

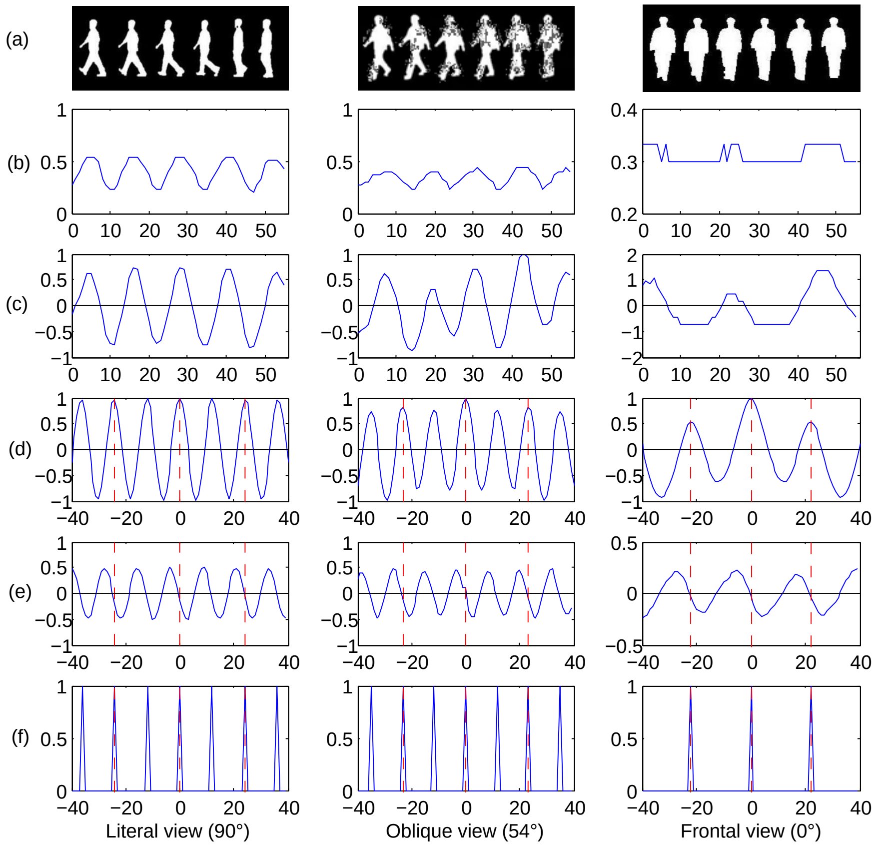


Figure 3.1: *Gait period* analysis. (a) Sample gait silhouettes under various views. (b) Aspect ratio of silhouette bounding box. (c) Normalization of aspect ratio followed by mean filtering. (d) Autocorrelation of aspect ratio under various offset. (e) First order derivative based on (d). (f) Peak positions indicating the periods.

transition position is defined at the zero crossing point along the positive-to-negative direction. As mentioned, because of the bilateral symmetry of human gait, the autocorrelation signals (under any view except approximate frontal view) contain minor peaks located half way between each pair of consecutive major peaks. So, the final period transition positions are shown and aligned using dash lines on Figure 3.1(f).

Here, the aspect ratio is used instead of the whole gait silhouette [98][99][100][101] in the autocorrelation because it can significantly reduce computational complexity for the period analysis. This advantage will be very helpful for real-time application because the *gait period* estimation must be repeatedly applied to every gait sequence.

3.3 Weighted Binary Pattern (WBP)

In combination of Sections 3.3.1, 3.3.2 and 3.3.3, a novel *Weighted Binary Pattern (WBP)* will be extracted from the synthetic template i.e. *GEI*.

3.3.1 Partial local binary pattern based features

Partial ideas of *Local Binary Pattern (LBP)* [65][66] are adopted and further adjusted to generate a binary sequence from *GEI* for each individual gait sequence. The binary sequence will be further developed to become a kind of gait feature.

LBP is derived from a general definition of texture in a local neighborhood. In this study, it describes features of the targeted texture image (i.e. *GEI*) where human gait frequency, local spatial-temporal gait information, and global shape of human body are embedded. *LBP* is made invariant against the rotation of the image domain, and supplemented with a rotation invariant measure of local contrast. In this way, *LBP* can be applied on *GEI* to make it more robust against uncertainties of gait information due to changes of walking conditions.

In the proposed method, *GEI* is first constructed from a sequence of aligned gait silhouettes within complete *walking cycle*(s) (see Section 2.1). Then, the *LBP* operator (see Section 2.2) is applied to each b^{th} block (e.g. 3×3 pixels) on *GEI* to generate a binary number (e.g. 8-digit binary number), denoted as s_b. The first block starts at top-left corner of *GEI*, then is shifted right or down by 1 pixel at a time to cover the whole *GEI*. The same order of blocks is used for every *GEI*.

The original *LBP* uses a histogram to collect local statistics of the binary pattern (s_b) in each image block where *LBP* is carried on. However, we found that the histogram also misses out the detailed local texture information although it well demonstrates the local statistics. Thus, we do not use the histogram as a common *LBP*. Instead, to preserve the more local information as much as possible, a sequence of the binary numbers ($S = \{s_b\}_{b=1}^{N_B}$) is directly used as a gait feature, where N_B is the total number of blocks on *GEI*. When the size of each image block is $B \times B$ and the size of each *GEI* is $M \times N$, $N_B = (M-B+1)\times(N-B+1)$.

The difficulty is such kind of binary pattern (S) contains a large number of bits. Treating all bits equally is not efficient and unnecessary. Hence, the weighting technique is proposed in Section 3.3.2 to discriminate significances of bits. The weight is calculated on each bit according to its ability of intra-person and inter-person discriminability.

3.3.2 Variance bit-weighting analysis

From Section 3.3.1, the binary pattern (S) is extracted from *GEI* as a primary gait feature. This section describes an adaptive variance bit-weighting technique to discriminate significance of each bit in S. This process will give a final gait feature, called *WBP*, which is defined as S plus weight values. However, in this section, the gait feature is referred to the primary gait feature S.

The bit with higher weight value should be the bit which can better recognize human gait. Thus, the question now is to have a way to measure the importance of individual bit. We borrow the idea of information measurement using entropy. That is, variables containing rich information usually have more uncertainty which can be inferred from the corresponding variance. Moreover, the high significant bits should be able to maintain consistency of gait features between same subjects (i.e. intra-person) and also conserve differentiability between different subjects (i.e. inter-person).

Thus, the following two steps are proposed to calculate the weight values for the purposes as mentioned above. A training dataset is given as $\{G_j^k \mid 1 \leq j \leq N, 1 \leq k \leq N_j\}$, where G_j^k is a gait sample k of subject j, N is the total number of training subjects, N_j is the total number of training samples of subject j, and different gait samples of the same subject may be recorded from different walking conditions. The method proposed in Section 3.3.1 is used to extract a primary gait feature S_j^k from a gait sample G_j^k.

Step1: To calculate intra-person information within gait features from same subject

For each subject j, there are multiple gait features S_j^k extracted from multiple walking video sequences G_j^k in the training dataset. Thus, it requires to generate one representative gait feature $\overline{S}$ along with its intra-person information for each individual subject. The representative gait feature for subject j ($\overline{S}_j$) is estimated as follow.

$$\overline{S}_j(i) = \left| \frac{\sum_{k=1}^{N_j} S_j^k(i)}{N_j} \right|_I \tag{3.2}$$

where $|a|_I = 0$ if $a < 0.5$, otherwise $|a|_I = 1$, $S_j^k(i)$ is the bit i of S_j^k, and $\overline{S}_j(i)$ is the bit i of $\overline{S}_j$. In other words, the bit i of the representative gait feature for subject j is '0' if the majority of bit-i values from training gait features of the subject j is '0', otherwise it will be assigned to '1'.

Since each bit i in the representative gait feature of subject j is determined from multiple gait features of subject j, therefore we can calculate disperse or invariance of variable X_{ij}, where X_{ij} contains bit-i values from all gait features of subject j in the training dataset. After the mean (μ_{ij}) and the variance (σ_{ij}) of the variable X_{ij} is obtained, Equation (3.3) calculates the invariant intra-person information for the bit i of the representative gait feature of subject j.

$$p_{ij} = 1 - \sqrt{\sigma_{ij}} \tag{3.3}$$

where μ_{ij} and σ_{ij} are calculated as follows.

$$\mu_{ij} = \frac{\sum_{k=1}^{N_j} S_j^k(i)}{N_j} \tag{3.4}$$

$$\sigma_{ij} = E[(X_{ij} - \mu_{ij})^2] = \frac{1}{N_j} \sum_{k=1}^{N_j} (S_j^k(i) - \mu_{ij})^2 \tag{3.5}$$

where E is the expected value.

At this stage, p_{ij} gives a quality of the bit i for describing subject j.

Step2: To calculate inter-person information between representative gait features from different subjects

Once we get the representative gait feature along with its intra-person information for each subject, then inter-person information for the bit i from N different subjects is estimated as below.

$$w_i = \sqrt{\frac{\sum_{j=1}^{N} p_{ij}(\overline{S}_j(i) - \mu_i)^2}{N}} \tag{3.6}$$

where p_{ij} is calculated from Equation (3.3), $\overline{S}_j(i)$ is the bit-i value from the representative gait feature of subject j, and μ_i is the mean value of $\overline{S}_j(i)$, $j = 1$, ..., N.

The value calculated in Equation (3.6) is used as the bit weight. w_i gives a quality of the bit i for distinguishing different subjects.

3.3.3 Similarity measurement

A weighted distance is calculated as below to measure gait similarity.

$$d = \sum_{i=1}^{N_I} w_i |S_1(i) - S_2(i)| \tag{3.7}$$

where d is a distance between two gait features (S_1 and S_2), w_i is calculated from Equation (3.6), and N_I is the total number of bits in the gait feature. The smaller value of d, the more possibility that S_1 and S_2 belong to the same subject.

3.4 Experiments

The proposed gait recognition method has been tested on three well-known gait databases, the CASIA gait database A, the CMU Mobo gait database, and the CASIA gait database C. The brief explanation is given below in each subsection. We manage our purposed method into two layers. At the first layer, only the binary patterns generated from Section 3.3.1, called Partial-*Local Binary Pattern* (P-*LBP*), are used to generate gait features. At the second layer, the variance bit weighting scheme described in Section 3.3.2 is applied to binary patterns resulted from Section 3.3.1 to generate gait features, known as *Weighted Binary Pattern (WBP)*.

Table 3.1: Experimental results on the CASIA gait database A.

Method	0°(%)	45°(%)	90°(%)
P-*LBP*(8,1)	92.50	86.25	86.25
WBP(8,1)	**100.00**	**100.00**	**98.75**
P-*LBP*(12,2)	87.50	90.00	83.75
WBP(12,2)	**97.50**	**97.50**	**92.50**
P-*LBP*(16,3)	87.50	83.75	**5.00**
WBP(16,3)	**97.50**	**97.50**	**5.00**
P-*LBP*(24,4)	87.50	**5.00**	**5.00**
WBP(24,4)	**95.00**	**5.00**	**5.00**

Regarding experimental validation, the leave-one-out cross validation is used for the case of fixed walking condition. On the other hand, for the case of across walking conditions, one walking condition is used for gallery data and another walking condition is used for probe data. In our experiments on both the CASIA gait database A and the CMU Mobo gait database, we show the results when using various parameter setups for P-*LBP* and *WBP*. In addition for the CASIA gait database C, Cummulative Match Scores (CMS) [30] are used to quantitively assess the recognition performance. The CMS value α corresponding to rank r indicates a fraction $100 \cdot \alpha$ % of probes whose top r matches must include the real identity matches.

3.4.1 Results on the CASIA gait database A

This database is used to testify the proposed gait feature based on different parameter setups of P-*LBP*(p,r) and *WBP*(p,r), where p is a number of neighboring pixels being considered and r is a radius from reference pixel (see Figure 2.3).

From Table 3.1, it is clearly seen that the parameter of (p,r)=(8,1) has the best performance for all walking directions. This is because the setup of (p,r)=(8,1)

Table 3.2: Comparison of rank 1 performance on the CASIA gait database A.

Method	0°(%)	45°(%)	90°(%)
BenAbdelkader et al. [35]	72.50	–	–
BenAbdelkader et al. [36]	82.50	–	–
Collins et al. [103]	71.25	–	–
Lee et al. [104]	87.50	–	–
Phillips et al. [105]	78.75	–	–
Wang et al. [106]	88.75	–	–
Su et al. [107]	89.31	–	–
Lu et al. [108]	82.50	–	–
Wu et al. [109]	90.00	90.00	83.00
Geng et al. [110]	90.00	95.00	90.00
The proposed method (*WBP*)	**100.00**	**100.00**	**98.75**

can capture the best local pattern information of human gait. When the radius r is increased, it still works fairly well for 0° but bad for 45° and 90°. This is because the oblique and frontal views contain less obvious gait identity information. Moreover, it can be seen that *WBP* can provide the better performance than P-*LBP* for all walking directions.

Moreover, from Table 3.2, it is clearly seen that the proposed *WBP*-based method outperforms other existing methods for all three views by achieving all final accuracies close to 100%.

3.4.2 Results on the CMU Mobo gait database

In this experiment, the parameter used for P-*LBP* and *WBP* is (p,r)=(8,1) and only lateral view is considered since it is the best case from the observation above. Table 3.3 demonstrates experimental results on the case of fixed walking condition. That is, both gallery data and probe data are under the same walking

Table 3.3: Experimental results on the CMU Mobo gait database for the case of fixed walking condition.

Walking condition	P-*LBP*	*WBP*
Slow walk (fs)	**100.00**	**100.00**
Fast walk (fq)	**100.00**	**100.00**
Slow walk with a ball (fb)	98.67	**100.00**
Slow walk at a fixed slope (fi)	98.00	**98.67**

Table 3.4: Experimental results on the CMU Mobo gait database for the case of across walking conditions.

Gallery	Probe	P-*LBP*	*WBP*
fs	fq	88.00	**92.00**
fq	fs	86.67	**92.00**
fs	fb	64.67	**72.67**
fb	fs	72.00	**74.67**
fq	fb	58.00	**60.67**
fb	fq	58.00	**63.33**

conditions. Table 3.4 demonstrates experimental results on the case of across walking conditions. That is, gallery data and probe data are under the different walking conditions.

As shown in Table 3.3, the proposed methods including P-*LBP* and *WBP* achieve very good performance on the CMU Mobo gait database for the case of fixed walking condition. In addition, from Table 3.4, *WBP* can significantly outperform P-*LBP* in the case of across walking conditions.

Moreover, the proposed *WBP*-based method is compared with other existing methods which use the same CMU Mobo gait database, as shown in Table 3.5. It can be seen that the performance of our proposed method still can be ranked

Table 3.5: Comparison of rank 1 performance on the CMU Mobo gait database.

Gallery	Probe	Chai et al. [37]	Shi et al. [111]	Tan et al. [30]	*WBP*
fs	fs	84.00	93.76	**100.00**	**100.00**
fq	fq	73.00	96.53	**100.00**	**100.00**
fb	fb	–	91.68	96.00	**100.00**
fi	fi	–	90.81	–	**98.67**
fs	fq	–	–	**96.00**	92.00
fq	fs	–	–	**92.00**	**92.00**

Table 3.6: Twelve experiments on the CASIA gait database C for the case of across walking conditions, where fs is slow walk, fn is normal walk, fq is fast walk, and fb is normal walk with a bag.

Exp.	A	B	C	D	E	F	G	H	I	J	K	L
Gallery	fs	fn	fq	fn	fb	fn	fb	fs	fb	fq	fq	fs
Probe	fn	fs	fn	fq	fn	fb	fs	fb	fq	fb	fs	fq

on the top.

3.4.3 Results on the CASIA gait database C

In this section, the experiments are carried out for both cases of fixed and across walking conditions as above. To simplify the explanation, our experiments for the case of across walking conditions are categorized into 12 small tests which are labeled as A, B, C, ..., K, L (see the details in Table 3.6). The recognition performances (i.e. CMS curves) of the proposed methods are shown in Figures 3.2 and 3.3.

As shown in Figure 3.2, the proposed *WBP* achieves very high accuracy for the case of fixed walking condition. In Figure 3.3, for the case of across walking conditions, the high performances are obtained for the tests A, B, C, D, E, and

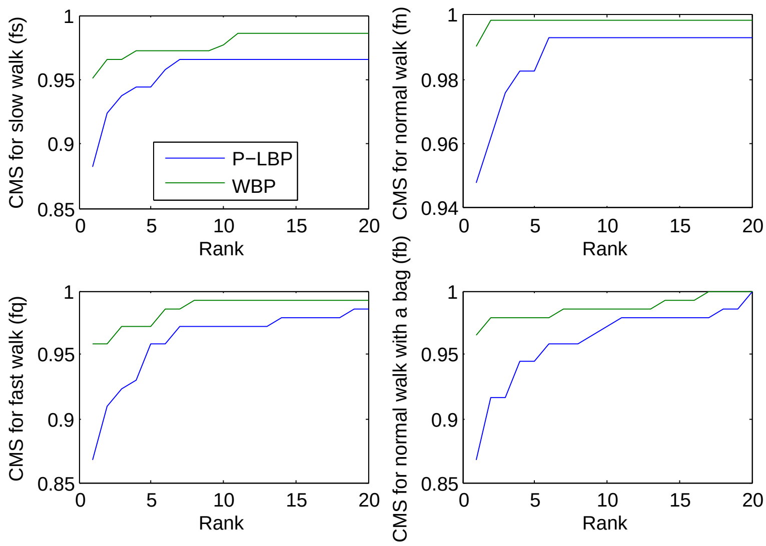


Figure 3.2: CMS curves on the CASIA gait database C for the case of fixed walking condition.

F (see descriptions of the labels in Table 3.6). This is because, for the tests A, B, C and D, the normal walking pattern is still quite closed to the slow and fast walking patterns. For the tests E and F, both walking conditions share the same normal walking pattern. The only difference is that one is with a bag and another is without a bag.

The moderate performances are achieved for the tests G, H, I, and J. In these tests, there are changes in two walking conditions including carrying/not carrying a bag and walking speed. The proposed *WBP* can still achieve fairly good results. However, the results for the tests K and L are not very good because large speed changes can significantly alter individual walking patterns.

Moreover, Table 3.7 shows comparison between the proposed method (*WBP*) and other existing methods on the same CASIA gait database C. It can be seen that the proposed method achieves the highest performance, especially in the last case of (fn, fb).

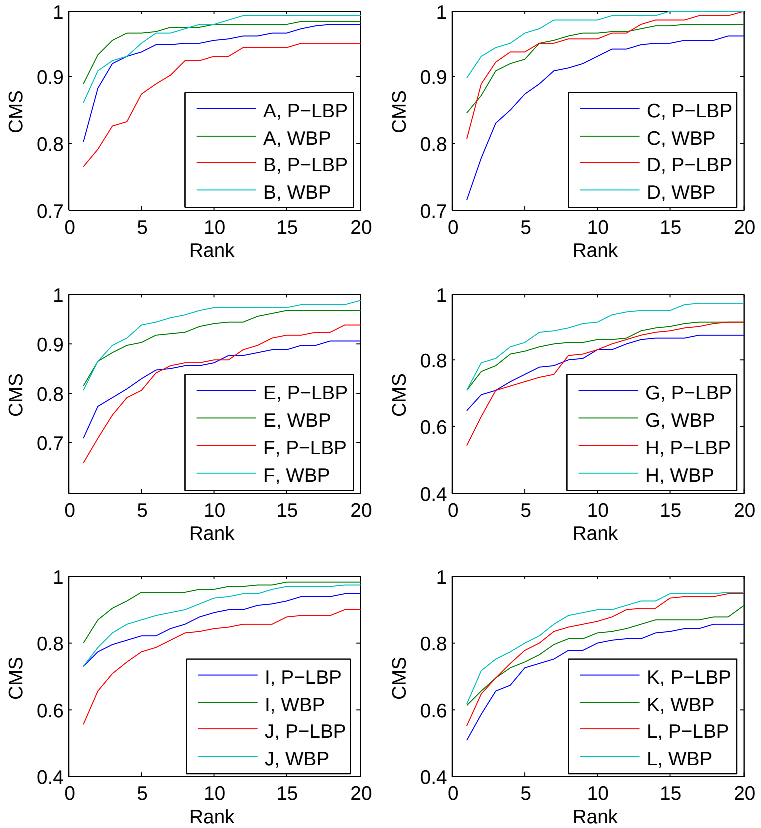


Figure 3.3: CMS curves on the CASIA gait database C for the case of across walking conditions.

3.5 Conclusion

This chapter has proposed a new method to extract a robust gait feature, *Weighted Binary Pattern (WBP)*, from a synthetic template (i.e. *GEI*) of a video sequence of walking human. To build up *WBP*, partial ideas of *LBP* are adopted and adjusted to construct the binary pattern, in stead of the histogram, on *GEI* extracted from a sequence of aligned silhouettes. Then, the variance bit weighting is applied to such binary pattern to discriminate the significance of each bit in the gait feature. In other words, less important and redundant bits are taken off. The extensive experiments on the three well-known gait databases show that the proposed method outperforms other existing methods in the literature.

In addition, the proposed method can be further improved in three aspects.

Table 3.7: Comparison of rank 1 performance on the CASIA gait database C.

(Gallery , Probe)	(fn , fn)	(fn , fs)	(fn , fq)	(fn , fb)
GEI [27]	96	74	83	50
Uniprojective [30]	97	84	88	36
HTI [56]	94	85	88	51
Wavelet packet [112]	93	83	85	-
Orthogonal projections [113]	98	80	80	18
NDDP [114]	97	85	74	16
Gait curves [115]	91	65	70	26
The proposed method (*WBP*)	**99**	**86**	**90**	**60**

First, the pre-processing stage can be introduced to select noise-free frames from a video sequence to generate the synthetic template. Second, multiple synthetic templates can be generated for each *walking cycle* rather than only one synthetic template generated in this study. Third, a non-linear binary pattern comparison technique will be added for an improvement on the similarity measurement. These aspects will be considered in the future work.

NOTE: Although *GEI* is shown to be less robust than the proposed *WBP*, it is still used as a baseline gait feature in our proposed methods in Chapters 3 and 4. This is because *GEI* contains very rich gait information preserving a spatial representation of human shape. The rich content of *GEI* will provide substantial correlations across views/speeds which are naturally bridged by the spatial information.

Chapter 4

Gait Recognition under Various Views based on View Transformation Model

4.1 Introduction

The aim of this study is to develop robust gait recognition which can tolerate view changes and to discuss other related challenges including automatic view detection and view optimization for building up *VTM*. A novel method is proposed to solve the problem of view changes by using *VTM*. It aims to learn a mapping relationship between gait features observed across views. In principle, *VTM* is used to transform gait feature from one view (source) into another view (target). In this way, gait features from various views can be transformed into a common view using trained *VTM*s. Then, gait similarity measurement can be carried out without difficulty.

Moreover, trained *VTM*s will be generic to any subject. The dataset used for constructing *VTM*s can be independent from the dataset used for recognizing gaits. To construct a more accurate *VTM*, it is practical to invite enough experienced actors to perform the required walking actions. In this way, precise gait data can be obtained from many subjects and on every required view for *VTM* construction. *VTM*s can cover a very dense view range such as ($0°$, $10°$, $20°$, ..., $180°$). In the recognition phase, depending on a detected view of probe data, corresponding trained *VTM* can normalize its view onto any other view to match the data in gallery dataset.

In this chapter, the methods are proposed based on two types of *VTM*, namely *dVTM* and *rVTM*. *dVTM* is established through a matrix factorization process. *SVD* is adopted to decompose the gait matrix into a view-independent matrix and a subject-independent matrix which is used further to construct *VTM*s. The gait matrix is created using the training dataset such that each row contains gait information from the same view but different subjects, and each column contains gait information from the same subject but different views. The proposed method constructs *dVTM* based on *GEI* optimized by *LDA*. The truncated *SVD* is also introduced to avoid oversizing and overfitting of *dVTM*. However, the performance of *dVTM* is limited by its linear processes on global features.

To improve the recognition performance, a new method is proposed to reformulate *VTM* construction as a regression problem based on carefully selected local motion features, called *rVTM*. Instead of linear processing carried out by *dVTM*, *rVTM* follows a non-linear approach to build up a more efficient *VTM*. The regression concept is used to reveal correlated motions of gaits from different views. *rVTM* consists of multiple regression processes which are trained to estimate gait feature under one view using correlated information in gait feature(s) under other view(s). The local correlated features, called *Region of Interest (ROI)*, under shared motions are exposed using rough body segmentations and correlation coefficients.

Among three adopted regression techniques (i.e. *Multi-Layer Perceptron (MLP)*, *Support Vector Regression (SVR)*, and *Sparse Regression (SR)*), *SR* can achieve the highest performance. This is because *SR* can further refine *ROI* due to its unique sparsity properties such that it may reduce the data redundancy in *ROI* significantly. In detail, *ROI* is comprised of a group of pixels on *GEI* and each pixel is selected independently from the source gait feature to ensure only a high correlation with the target gait feature. Thus, pixels of *ROI* can be selected from overlapped motion areas due to overfitting in the training process. In that case, these pixels will be repeatedly used in the regression process. This is known as data redundancy in *ROI*.

To address this problem, *SR* based on the elastic net is adopted to reduce the possible data redundancy. *ROI* will be further selected for the second time throughout the training process of the *sparse regression*. This refined *ROI* will contain significantly reduced redundant information and will be helpful in obtaining stable and non-overfitting regression models in *rVTM* construction. Our experimental results have also indicated that the proposed *rVTM*-based method outperforms the state-of-the-arts for gait recognition under various views.

The rest of this chapter is organized as follows. Problem formulation is analyzed in Section 4.2. Pre-processing is explained in Section 4.3, followed by adopted gait features discussed in Section 4.4. *dVTM* and *rVTM* constructions are proposed in Sections 4.5 and 4.6 respectively. Gait similarity measurement is given in Section 4.7. Implementations of gait recognition under various views for different practical cases are described in Section 4.8. Experimental results are shown in Section 4.9 and conclusions are drawn in Section 4.10.

4.2 Problem Formulation

In this section, *dVTM* and *rVTM* are formulated in their general forms. The advantages of *rVTM* are also discussed over *dVTM*. In this study, *GEI* [27] is adopted to represent global gait information under complete walking period(s). *GEI* is an appearance-based gait feature which has been proved to be very robust and efficient for gait recognition [27][64]. However, it contains high view-dependent information. In order to achieve the efficient gait recognition system which is also robust to view change, the proposed view-invariant system is developed based on *GEI* using the concept of generic *VTM*. As mentioned in the introduction, *VTM* is generic because it is trained using the independent dataset and can be applied for view normalization of any known/unknown subject. Moreover, view change is a kind of external factor which will affect the shooting conditions but is independent to subjects being shot. Thus, it has the similar impacts to all subjects.

dVTM is a conventional approach [52] for *VTM* construction based on a matrix decomposition concept. This method solves the view transformation problem via the equivalent decomposition problem. In this method, the gait matrix

$A \in \Re^{N_v N \times N_s}$ is first built up such that each row contains gait features from the same view of different N_s subjects and each column contains gait features from the same subject under different N_v views, where N is the dimension of gait feature. Then, SVD (see Section 2.4) decomposes A into a subject-independent matrix (P) and a view-independent matrix (V) such that $A = USV^T = PV^T$ where $U \in \Re^{N_v N \times N_s}$, $S \in \Re^{N_s \times N_s}$, $V \in \Re^{N_s \times N_s}$, $P = [P_{\theta_1}, ..., P_{\theta_{N_v}}]^T$ and $V = [V_1, ..., V_{N_s}]^T$. Therefore, gait features of subject m can be factorized into two components as in Equation (4.1).

$$\begin{aligned} G_{m,\theta_i} &= P_{\theta_i} V_m \\ G_{m,\theta_k} &= P_{\theta_k} V_m \end{aligned} \tag{4.1}$$

where G_{m,θ_i} and G_{m,θ_k} are GEIs of subject m from θ_i and θ_k respectively, V_m is the intrinsic vector of subject m, and P_{θ_i} and P_{θ_k} are projection matrices to transform the intrinsic vector to gait features under θ_i and θ_k respectively. Thus, given GEI of any subject from view θ_i (G_{θ_i}), it can be transformed into view θ_k as below.

$$G_{\theta_k} = P_{\theta_k} P_{\theta_i}^{-1} G_{\theta_i} \tag{4.2}$$

where $P_{\theta_i}^{-1}$ is the pseudo inverse of P_{θ_i}. However, $dVTM$ still has many constraints as mentioned in the introduction. To tackle these constraints, view transformation is completely reformulated into a problem of regression (i.e. $rVTM$).

The basic idea to achieve *cross-view gait recognition* based on $rVTM$ is briefly summarized here. First, GEIs from different views of a person share common features. Second, a pixel in GEI is highly correlated to its neighboring pixels in the area of shared motion. Such similarity and embedded relations of the gait features from different views can be revealed using regression. Inspiring by these rationales, a group of pixels in GEI, namely *Region of Interest (ROI)*, from one view can be used to estimate the corresponding correlated pixel in GEI from another view.

Based on the proposed $rVTM$, Equations (4.3) and (4.4) present the idea that uses the regression procedure to predict a pixel in GEI under one view from the corresponding selected *ROI* in *GEI* under another view.

$$
\begin{aligned}
G_{\theta_k} &= f(\hat{W}, G_{\theta_i}) \\
\hat{W} &= \underset{W}{\operatorname{argmin}} \; ||G_{\theta_k} - f(W, G_{\theta_i})||_2
\end{aligned} \tag{4.3}
$$

where f is a function that contains well refined regression processes, W is a weight vector, $G_{\theta_k} = \{g^n_{\theta_k}\}^N_{n=1}$, $G_{\theta_i} = \{g^n_{\theta_i}\}^N_{n=1}$, $g^n_{\theta_k}$ and $g^n_{\theta_i}$ are the n^{th} pixels in G_{θ_k} and G_{θ_i} respectively, N is the total number of pixels in *GEI*, and $|| \cdot ||_2$ is L_2 norm. Then, Equation (4.3) can be expressed as follow.

$$
\begin{aligned}
\begin{bmatrix} g^1_{\theta_k} \\ g^2_{\theta_k} \\ \cdot \\ \cdot \\ \cdot \\ g^N_{\theta_k} \end{bmatrix} &= \begin{bmatrix} f_1\left(\hat{W}_1, ROI^1_{\theta_i}\right) \\ f_2\left(\hat{W}_2, ROI^2_{\theta_i}\right) \\ \cdot \\ \cdot \\ \cdot \\ f_N\left(\hat{W}_N, ROI^N_{\theta_i}\right) \end{bmatrix} \\
\hat{W}_n &= \underset{W_n}{\operatorname{argmin}} \; ||g^n_{\theta_k} - f_n\left(W_n, ROI^n_{\theta_i}\right)||_2
\end{aligned} \tag{4.4}
$$

where f_n is a regression function, W_n is a weight vector, $ROI^n_{\theta_i} \subset G_{\theta_i}$, and each pixel in $ROI^n_{\theta_i}$ is highly correlated to the corresponding target pixel ($g^n_{\theta_k}$). The detailed explanation for selection of *ROI* and regression training process of f are given in the later sections of this chapter.

$rVTM$ contains several advantages over $dVTM$ as follows.

- Direct problem formulation: $rVTM$ directly adopts the regression concept to solve the view transformation problem. In contrast, $dVTM$ indirectly solves the view transformation problem. The SVD is first adopted to solve the equivalent decomposition problem. Then, its intermediate results is sequentially used to answer the view transformation problem. That is when $dVTM$ obtains the optimized solutions for the decomposition problem, it might not be optimal for the consequent view transformation problem.

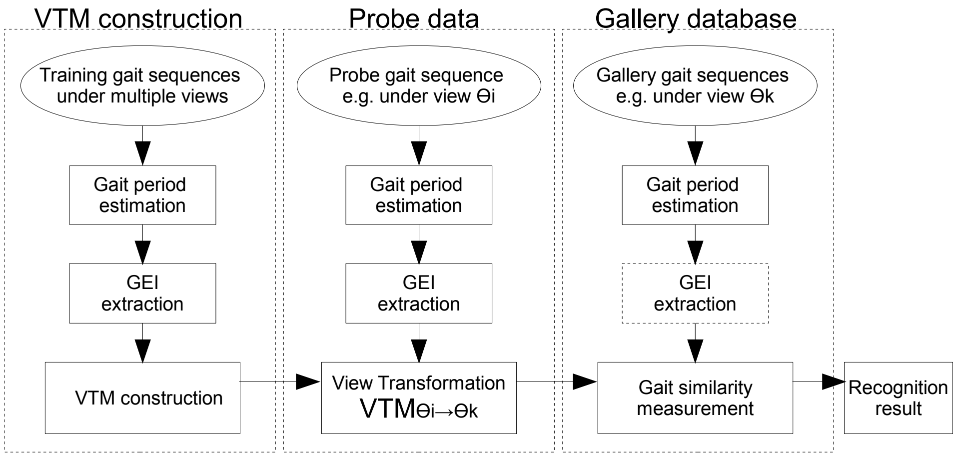


Figure 4.1: The proposed framework of *VTM*-based gait recognition.

- Intuitive method analysis: *rVTM* rationally solves the view transformation problem using a sensible objective by explicitly analyzing the problem based on existing correlated gait information. In contrast, *dVTM* solves the problem based on a vague problem statement. First, it can be argue that gait feature might not be able to be implicitly factorized into explicit two components. The latent structure of gait feature might be more complex. Second, from Equation (4.1), P and V are virtually defined for the decomposition analysis. There has been no proof to confirm the physical meaning of P and V as the view projection and the intrinsic gait feature respectively.

- Local feature selection: From Equation (4.4), the regression model f_i is trained based on the well selected *ROI* which is a local subset of a global feature *GEI*. In contrast, *dVTM* requires the global gait feature in the decomposition process. P and V from Equation (4.1) are formed as linear combinations of all pixels in *GEI* such that P and V are not sparse. Based on our study, the local feature can provide the better view transformation than the global feature which instead contain irrelevant and unreliable information for the transformation process.

- Nonlinear problem solving: The performance of *dVTM* is bounded by limitations of a linear process. In contrast, the performance of *rVTM* can be enhanced by arbitrary choices of nonlinear kernels.

Regarding the computational complexity, training *VTM* may take time but transformation process itself is very fast given a trained *VTM*. In our experiment, it took approximately 10 minutes to train one *VTM* but it took only less than 0.001 second to use *VTM* for one view transformation.

Figure 4.1 illustrates the proposed framework of *VTM*-based gait recognition. The processes for constructing *VTM* and building up the gallery database can be done offline. *Gait period* estimation and *GEI* extraction are explained in Sections 3.2 and 2.1 respectively. The key components of *VTM* construction will be explained in the rest of this chapter.

4.3 Pre-processing and View Classification

The pre-processing (e.g. *gait period* estimation) for the standard gait recognition is explained in Section 3.2. In addition, this section gives an explanation of additional pre-processing (i.e. *view classification*) for gait recognition under various views.

As mentioned in the related works (see Section 1.2.2.3), the methods of the third category (which include our proposed *VTM*-based methods) require the knowledge of view for each gait sequence before applying the corresponding *view transformation model*. To this end, a view estimation step is necessary and proposed. This step is called "automatic *view classification*". The term of "classification" is used because, in our design, views are discretized into a finite number of classes. In the previous work [48], *view classification* methods have been introduced. However, *view classification* with high performance is still an open challenge.

Given a gait feature $G \in \Re^N$ and a pre-defined similarity measurement function d, the corresponding view for G can be determined by $\theta = \operatorname*{argmin}_{\theta_i, 1 \le i \le N_v} d(G, \mu_i)$ where μ_i is the representative gait feature under view θ_i. In this study, instead of calculating the similarity between G and μ_i directly in the gait feature space $\Re^N$, they are transformed into a more robust feature space before the measurement. The proposed method projects gait features into *LDA* in *PCA* transformed space [116]. Compared with *LDA*, *LDA* in *PCA* transformed space is more robust to outliers, which combines the benefits of both *PCA* and *LDA*. Moreover, this complete *LDA* in *PCA* transformed space can be more efficient than the straightforward strategy of *PCA* plus *LDA* [117][118] because it can prevent losing discriminative information in *PCA* step. The main idea of *LDA* in *PCA* transformed space is briefly summarized below. The detailed process can be referred to [116].

4.3.1 LDA in PCA transformed space

In the proposed *view classification*, the gait feature space $\Re^N$ is transformed into the *LDA* in *PCA* transformed space $\Re^{l+d}$ using a projection matrix Φ. In this particular transformation, $\Phi = W_m^T Y$ where $W_m \in \Re^{m \times N}$ is a *PCA* projection matrix which is constructed based on a training dataset ($\mathbb{G} = \{G_s\}_{s=1}^{N_s}$ containing N_s samples of gait features from N_v views) and $Y = Y_1 : Y_2$ ($Y \in \Re^{m \times (l+d)}$, $Y_1 \in \Re^{m \times l}$, $Y_2 \in \Re^{m \times d}$) is a matrix containing $l+d$ Fisher optimal discriminant vectors which are calculated in *LDA* step.

To calculate Y [116], the gait feature space $\Re^N$ is first transformed into the *PCA* space $\Re^m$ using W_m. In this *PCA* transformed space, the within-class scatter matrix is created and further split into its null space and its orthogonal complement. It can be verified that all discriminatory information with respect to Fisher criterion is contained in these two subspaces. Based on a revised Fisher criterion in [116], the isomorphic mapping technique is employed for the calculation of the Fisher optimal discriminant vectors (Y) in these two subspaces where Y_1 is calculated in the null space and Y_2 is calculated in the orthogonal complement.

4.3.2 View classifier

Given $\theta_1, ..., \theta_{N_v}$ are N_v different views, view (θ) of each gait (G) can be determined as follow.

$$\theta = \underset{\theta_i, 1 \leq i \leq N_v}{\operatorname{argmin}} \, d(\Phi^T G, \Phi^T \mu_i) \tag{4.5}$$

where μ_i is the mean of training gait features under view θ_i and d is a chosen similarity measurement function e.g. Euclidean distance [119]. Compared with the method for *view classification* based on GP or SVM in [48], our method is rather simple, lower computational complexity, and more robust to the outliers. Experiments to verify efficiency of the proposed view classifier will be carried out in Section 4.9.

4.4 Gait Feature Preparation

GEI (as explained in Section 2.1) is adopted as a baseline gait feature in this study. For *dVTM*, *LDA* is applied on *GEI* to acquire an optimized version of the feature. There are two main advantages of using *LDA*. The first benefit is to significantly reduce dimension of gait feature, which is very useful aspect in real-time applications. The second benefit is to maximize a margin between gait features from different subjects and as well to minimize a distance of gait features that belong to the same subject.

As mentioned above, by applying *LDA* on *GEI*, it can achieve the reduction of impacts from gait feature's geometric properties and gait feature's dimension. Therefore, the optimized *GEI* is expected to be better factorized in the process of *dVTM* construction, being compared to the original spatial-domain *GEI*. *GEI* can be projected into *LDA* subspace before efficiently used in *dVTM* construction. This is because *dVTM* is built up mainly based on statistical analysis relying on mathematical concept, without using any sensible context in the feature space.

In contrast, the original *GEI* is used in *rVTM* construction. This is because spatial gait information in the feature space is necessarily required to build up *rVTM*. Such information provides correlated features across views which will be used for bridging gaps between gaits from different views. For more detail, a correlated motion analysis on *GEI*s under various views is discussed below.

It can be seen from Figure 2.1 that *GEI*s under various views are different. However, they are captured from the same 3D gait of an individual. Therefore, they must contain common information especially for the case of small view difference. This section observes correlations between *GEI*s from different views based on correlation coefficients [120]. Refer to each row in Figure 4.2, one pixel (p) in *GEI* from the target view is selected as an example, then the correlation coefficient is calculated between p and each pixel in *GEI* from the source view.

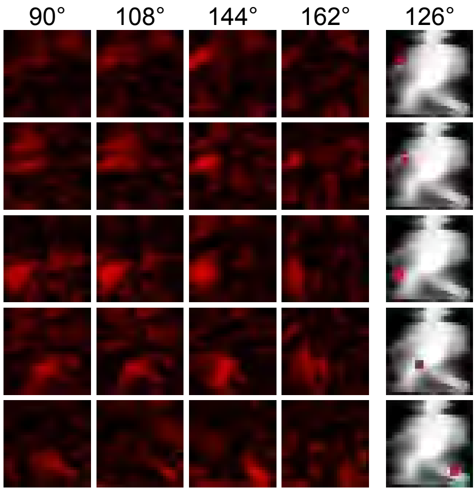


Figure 4.2: Correlated motions between *GEI*s from different views. The darker red color represents the higher correlation. The first four columns present correlation coefficients between each pixel in *GEI*s from various source views (90°,108°,144°,162°) and selected pixel in *GEI* from the target view (126°) shown in the last column. Each row corresponds to different target pixels (red pixels in the last column).

From Figure 4.2, several important points can be observed. First, source pixels which are close to the location of corresponding target pixel, usually have higher correlation (presented by darker red color). Second, based on a set of training data, a target pixel in one body part (e.g. leg) may be fairly correlated to source pixels from other body parts (e.g. arm). Such correlated information is not reliable because: 1) it is a weak correlation, for example, arms' swing can be naturally related to legs' movement; 2) it is not generic. For example, different persons may present various relationships between his arms' and legs' movements. Therefore, when this kind of correlated information is used in a regression process, it may lead to overfitting and unstable regression model for our *rVTM* construction. In this study, this problem will be addressed using rough body segmentations.

It is also observed that *GEI*s from closer views shares stronger correlations. *GEI*s from different source views provide various information correlated to *GEI* from the target view. Such that, we propose a more reliable view transformation to predict *GEI* from one view by using *GEI*s from multiple views (so called multi-view to one-view $rVTM$).

From the analysis, it can be seen that there is a kind of relation existing between different gait pixels under different views. Given that such relation can be established through a regression process, a new $rVTM$ can be possibly constructed.

4.5 Solution based on dVTM

This section describes $dVTM$ construction using a Truncated *Singular Value Decomposition* (TSVD) technique. $dVTM$ is developed based on GEI optimized by LDA as a gait feature.

In the proposed method, a gait matrix A is first created from the training dataset, as the left-hand side matrix in Equation (4.6). Each row contains gait information under the same view but from N_s different subjects. Each column contains gait information from the same subject but under N_v different views. Then, SVD factorizes the gait matrix A into a subject-independent matrix P and a view-independent matrix V as follow.

$$A = \begin{bmatrix} G_{1,\theta_1} & ... & G_{N_s,\theta_1} \\ . & . & . \\ . & . & . \\ . & . & . \\ G_{1,\theta_{N_v}} & ... & G_{N_s,\theta_{N_v}} \end{bmatrix} = USV^T = \begin{bmatrix} P_{\theta_1} \\ . \\ . \\ . \\ P_{\theta_{N_v}} \end{bmatrix} \begin{bmatrix} v^1 & ... & v^{N_s} \end{bmatrix} \tag{4.6}$$

where $G_{k,\theta_i}(\ 1 \leq k \leq N_s,\ 1 \leq i \leq N_v)$ denotes the gait feature of subject k under view θ_i. U is the $N_vN_g \times N_s$ orthogonal matrix where N_g is the dimension of the gait feature. V is the $N_s \times N_s$ orthogonal matrix. S is the $N_s \times N_s$ diagonal matrix containing the singular values. $P = [P_1, ..., P_{\theta_{N_v}}]^T = US$ where P_i is the $N_g \times N_s$ sub-matrix of US. v^k is the N_s dimensional column vector of V.

A vector v^k is an intrinsic gait feature of subject k for any view. P_{θ_i} is a projection matrix which can project an intrinsic vector v of any subject to the gait feature vector under a specific view θ_i. Thus, gait feature can be written in factorized form as follow.

$$G_{k,\theta_i} = P_{\theta_i} v^k \tag{4.7}$$

The subject-independent matrix P is used to construct $dVTM$ in common for any seen or unseen subject. For example, gait feature transformation from view θ_i to θ_j is obtained as below.

$$\begin{aligned} G_{k,\theta_j} &= P_{\theta_j} v^k \\ &= P_{\theta_j}(P^+_{\theta_i} P_{\theta_i}) v^k \\ &= P_{\theta_j} P^+_{\theta_i}(P_{\theta_i} v^k) \\ &= P_{\theta_j} P^+_{\theta_i} G_{k,\theta_i} \end{aligned} \tag{4.8}$$

where $P^+_{\theta_i}$ is the pseudo inverse matrix of P_{θ_i}.

Thus, $dVTM_{\theta_i \to \theta_j}$ that is used to transform gait feature from view θ_i to θ_j is constructed as in Equation (4.9).

$$dVTM_{\theta_i \to \theta_j} = P_{\theta_j} P^+_{\theta_i} \tag{4.9}$$

In order to simplify SVD computational complexity and make it less sensitive to trivial elements, a reduced version of SVD is adopted. There are three main advantages of using a reduced version of SVD. First, it reduces the dimension of $dVTM$. Therefore, it makes the algorithm perform faster. Second, it is more economic for storage. Third, it improves the accuracy of view transformation process because it avoids over-fitting problem in some how by removing the less important elements from the transformation model. These three benefits become significant when there are sufficient training data for constructing $dVTM$.

There are several reduced SVD solutions. TSVD [121] is an efficient one and selected in this study. In Equation (4.6), it can be seen that S is a diagonal matrix containing the singular values as follows.

$$S = \begin{bmatrix} \sigma_1 & & & & & & \\ & \sigma_2 & & & & & \\ & & \sigma_3 & & & & \\ & & & \cdot & & & \\ & & & & \cdot & & \\ & & & & & \cdot & \\ & & & & & & \sigma_{N_s} \end{bmatrix} \tag{4.10}$$

where $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_{N_s} \geq 0$ are the singular values of the matrix A. Its rank is N_s. Then, the reduced rank approximation for TSVD is by setting all but the first r ($r < N_s$) largest singular values to zero. Thus, the dimension of P_k in Equation (4.6) is reduced from $N_g \times N_s$ to $N_g \times r$. The other significant aspect is that such optimization avoids over-fitting problem by removing the less important elements from the transformation model.

4.6 Solution based on rVTM

As mentioned above in Section 4.2, $rVTM$ is built up from N regression processes, where N is the total number of pixels in *GEI*. Each regression process is trained to predict a pixel in *GEI* under target view using a relevant group of pixels or a *Region of Interest (ROI)* in *GEI* under source view. A regression function (f_n) in $rVTM_{\theta_i \to \theta_k}$ (see Equation (4.4)) is defined as follow.

$$\begin{aligned} \hat{g}^n_{\theta_k} &= f_n(\hat{W}_n, \gamma_n(G_{\theta_i})) \\ &= f_n(\hat{W}_n, ROI^n_{\theta_i}) \end{aligned} \tag{4.11}$$

where G_{θ_i} and G_{θ_k} are *GEI*s under views θ_i and θ_k respectively, $rVTM_{\theta_i \to \theta_k}$ is the transformation model used to transform G_{θ_i} to G_{θ_k}, $g^n_{\theta_k}$ is the n^{th} pixel in G_{θ_k}, $ROI^n_{\theta_i}$ is a group of pixels in G_{θ_i} for prediction of the corresponding target pixel $g^n_{\theta_k}$, and γ_n is *ROI* selection process. As similar to $dVTM$, to build up a reliable $rVTM$, a sufficient number of training pairs $(G_{\theta_i}, G_{\theta_k})$ from different subjects are required.

$rVTM$ can then be obtained through a regression (offline) training procedure based on an independent training dataset containing multiple subjects and sufficient various walking patterns from different views. In order to maintain the performance and reliability of $rVTM$, it can be continuously refined using online training processes. Moreover, in Equation (4.11), it can be seen that W and *ROI* (i.e. the key components of $rVTM$) are independent of subjects of interest. That is, $rVTM$ is obtained regardless of individual gait features. Thus, $rVTM$ can be used for view transformation for any seen or unseen subject.

As formulated in Equation (4.4), each regression is to train *ROI* in source *GEI* to predict the corresponding pixel in target *GEI*. Then, all estimated target pixels are merged to form a new *GEI* under the target view. Two important tasks, 1) *ROI* selection process (γ_n) and 2) regression training process (f_n), are clearly explained in the following subsections.

When the *ROI* selection and regression training process are successfully operated, then *rVTM* is obtained as a set of the trained regression processes as follow.

$$rVTM = \{f_n\}_{n=1}^{N} \tag{4.12}$$

In the testing phase, for example, $rVTM_{\theta_i \to \theta_k}$ can be used for view transformation as follow.

$$\begin{aligned} G_{\theta_k} &= G_{\theta_i} \odot rVTM_{\theta_i \to \theta_k} \\ &= \{W_n^T ROI_{\theta_i}^n\}_{n=1}^{N} \end{aligned} \tag{4.13}$$

where $ROI_{\theta_i}^n$ and W_n are obtained based on Equation (4.15) and Equation (4.17) or (4.18) or (4.21) respectively.

4.6.1 ROI selection

The *ROI* selection process contains two main steps: 1) locating a candidate area (A_c) based on a rough body segmentation and 2) using correlation strength to select elements of *ROI* from A_c. A_c is an area in source *GEI* which is likely to contain relevant gait pixels to the corresponding target pixel. Seeking *ROI* is performed in local region A_c instead of global *GEI*. In this way, predicting pixels in *GEI* under target view through *ROI* will be more reliable. Otherwise, it will be affected by noise such as shadows and analogous pixels which actually belong to motion in other body parts.

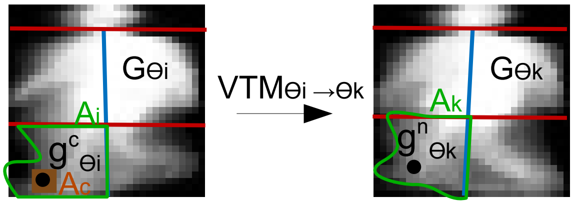


Figure 4.3: *ROI* selection. The blue lines are major axes. The red lines are body segmentations. The brown rectangle is a candidate area (A_c).

4.6.1.1 Step 1: Locating a candidate area (A_c) based on a rough body segmentation

In the first step to allocate A_c, *GEI* is first segmented into five regions by one vertical line and two horizontal lines as shown in Figure 4.3. To be different from torso and legs, head is not split because it looks more like a whole.

The vertical line is a major axis which vertically divides *GEI* into left and right sides. It is calculated from the dominant eigenvector of the covariance matrix of the tracked silhouette [43][122][123]. To make it clear, given $\{(x_i, y_i)|x_i \in \Re, y_i \in \Re, i = 1, ..., N_p\}$ denotes a set of 2D-coordinates of foreground pixels in the gait silhouette image, where N_p is the total number of foreground pixels in the gait silhouette image. The covariance matrix can be generated as follow.

$$\begin{bmatrix} COV(x,x) & COV(x,y) \\ COV(y,x) & COV(y,y) \end{bmatrix}_{2\times 2} \tag{4.14}$$

where $COV(x,y) = \sum_{i=1}^{N_p}((x_i - \overline{x})(y_i - \overline{y}))/N_p$ and $\overline{x}$ and $\overline{y}$ are the means of x and y, respectively. Thus, the major axis is the eigenvector of the covariance matrix corresponding to the larger eigenvalue. For each camera view, major axes calculated from different sample silhouettes can be slightly different. Thus, the average (in terms of the axis's slope) is used as a representative major axis of *GEI* under the camera view.

The horizontal lines are rough body segmentations which horizontally segment *GEI* into three parts, head (hair + face), upper body (torso + arms), and lower body (legs + feet). The portions of human body of *GEI*s under any two views can be significantly different when the two views are captured from different cameras with different parameters setting. The two horizontal lines are analyzed from the

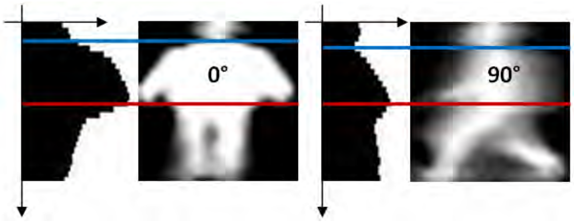


Figure 4.4: Projection histograms for body part segmentation process. The blue line is a border between head and upper body, and the red line is a border between upper body and lower body.

projection histogram of *GEI* on a direction of the major axis as shown in Figure 4.4. To do this, *GEI* is first projected on the major axis to create the histogram. Then, the histogram is smoothed by an average filter.

From Figure 4.4, a border between head and upper body is at the first saddle point from top to bottom or from head to feet. A border between upper body and lower body is claimed to be at the hip which is approximately located at the peak position of the histogram based on the observation (see Figure 4.4). This method is the rough body segmentation which does not require extremely high precision. It will be used to allocate the candidate area for *ROI* which will be then carefully filtered.

Once *GEI* is divided into five regions as shown in Figure 4.3, assume that the target pixel ($g^n_{\theta_k}$) is located in the region A_k: "Left-Lower body" region in G_{θ_k}, then the corresponding source pixel ($g^c_{\theta_i}$) must be located in the equivalent region A_i: "Left-Lower body" region in G_{θ_i}, where $g^n_{\theta_k}$ and $g^c_{\theta_i}$ pixels are assumed to correspond to the same 3D gait motion, but under different views θ_k and θ_i, respectively. Thus, a position of $g^c_{\theta_i}$ in A_i is estimated corresponding to the position of $g^n_{\theta_k}$ in A_k, which is proportional to sizes and rotations of the areas. A_c includes $g^c_{\theta_i}$ and its neighboring pixels.

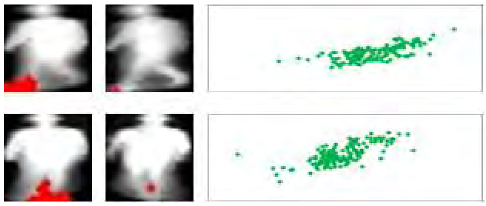


Figure 4.5: Relationship between source *ROI* and its corresponding target pixel. The first row is the *ROI* selection for $rVTM_{36^\circ \to 54^\circ}$ and the second row is the *ROI* selection for $rVTM_{18^\circ \to 162^\circ}$. For each row, the first image contains the allocated *ROI* (red pixels) for predicting the corresponding target pixel (red pixel) as shown in second image. The third image shows the relationship between the target pixel (y-axis) and the selected pixel from corresponding source *ROI* (x-axis) from various pairs of training samples $(G_{s,\theta_i}, G_{s,\theta_k})$.

4.6.1.2 Step 2: Using correlation strength to select elements of ROI from A_c

In the second step, *ROI*'s pixels are selected from A_c using correlation strength as follow.

$$ROI^n_{\theta_i} = \{p \ : \ |COR(p, g^n_{\theta_k})| > T\}, \quad p \in A_c \tag{4.15}$$

where T is the threshold and COR is the correlation coefficient [120]. Given a training dataset which contains N_s pairs of *GEI*s from θ_i and θ_k ($\{G_{s,\theta_i}, G_{s,\theta_k}\}^{N_s}_{s=1}$), $COR(g^m_{\theta_i}, g^n_{\theta_k})$ between the two pixel-variables is calculated as follow.

$$COR(g^m_{\theta_i}, g^n_{\theta_k}) = \frac{\sum^{N_s}_{s=1}(g^m_{s,\theta_i} - \bar{g}^m_{\theta_i})(g^n_{s,\theta_k} - \bar{g}^n_{\theta_k})}{\sqrt{\sum^{N_s}_{s=1}(g^m_{s,\theta_i} - \bar{g}^m_{\theta_i})^2}\sqrt{\sum^{N_s}_{s=1}(g^n_{s,\theta_k} - \bar{g}^n_{\theta_k})^2}} \tag{4.16}$$

where g^m_{s,θ_i} and g^n_{s,θ_k} are values of the m^{th} and n^{th} pixels in *GEI* training samples G_{s,θ_i} and G_{s,θ_k} respectively, $\bar{g}^m_{\theta_i} = \frac{\sum^{N_s}_{s=1} g^m_{s,\theta_i}}{N_s}$, and $\bar{g}^n_{\theta_k} = \frac{\sum^{N_s}_{s=1} g^n_{s,\theta_k}}{N_s}$.

The value of T is decided based on cross validation tests. This method provides the most relevant pixels ($ROI^n_{\theta_i}$) which have the closest motion-relation with the target pixel ($g^n_{\theta_k}$). Figure 4.5 shows examples of *ROI* selections based on the proposed method. It can be seen that regression model can be fitted well to the relationship between the target pixel and its corresponding *ROI*.

4.6.2 Regression training process

Three regression tools (*MLP*, *SVR*, *SR*) are employed as a case study. Given a training dataset of pairs $\{(G_{s,\theta_i,}, G_{s,\theta_k})\}_{s=1}^{N_s}$ where G_{s,θ_i} is *GEI* of training sample s from view θ_i, G_{s,θ_k} is *GEI* of training sample s from view θ_k, N_s is the total number of training samples, $G_\theta = \{g_\theta^n\}_{n=1}^N$, and N is the total number of pixels in *GEI*. The followings are explained based on this given training dataset.

4.6.2.1 Multi-layer perceptron (MLP)

MLP from Equation (2.9) is adopted to solve each model f_n from Equation (4.11) as follow.

$$\begin{aligned} g_{\theta_k}^n &= f_n(ROI_{\theta_i}^n) \\ &= \phi < W, (\phi < W^h, ROI_{\theta_i}^n > + b_h) > + b \\ &= \phi(\sum_{h=1}^{H} w_h(\phi(\sum_{p=1}^{P} w_p^h ROI_{\theta_i}^n(p)) + b_h)) + b \end{aligned} \tag{4.17}$$

where H is the total number of nodes in the hidden layer, P is the total number of pixels in *ROI*, and $ROI_{\theta_i}^n(p)$ is the pixel p in $ROI_{\theta_i}^n$.

For each regression process (f_n), three layers in *MLP* are defined [73][74] with *ROI* in source *GEI* as the input layer (P neurons), 20 neurons in the hidden layer, and the corresponding pixel in target *GEI* as the output layer (1 neuron). A sigmoid $\phi(d) = a/(1 + exp(-d)) - b$ is used as an activation function, where a and b are coefficients. In our study, a is 2 and b is 0.5. Hence, $\phi(d)$ is between -0.5 and 1.5 which covers the range of the *GEI*'s pixel value ([0,1]). In addition, the back propagation by Gradient Descent [73][74] is used to update weights in the training phase according to the desired values.

4.6.2.2 Support Vector Regression (SVR)

SVR from Equation (2.13) is adopted to solve each model f_n from equation (4.11) as follow.

$$\begin{aligned} g^n_{\theta_k} &= f_n(ROI^n_{\theta_i}) \\ &= \sum_{s=1}^{N_s}(\alpha_s - \alpha_s^*)K(ROI^n_{s,\theta_i}, ROI^n_{\theta_i}) + \ b \end{aligned} \tag{4.18}$$

where ROI^n_{s,θ_i} is the pixel values of $ROI^n_{\theta_i}$ from training sample s, and K is the kernel.

The three kernels (linear, polynomial, RBF) are evaluated in this study. Understandings of the parameters and the natures of the applications and the training data, highly assist in selecting the SVR's parameters. However, the parameters are usually selected by users based on a priori knowledge and/or an user expertise [124]. In this study, the following methods are used to estimate the initial values for the parameters. The optimal parameter set is then found simultaneously by minimizing the prediction error on the validation dataset.

The first main parameter is ϵ or the width of the tube for SVR as shown in Figure 2.5. To obtain an optimized solution, the ϵ may vary for each $rVTM$ construction of its particular pair of views. The value of ϵ should be proportional to the noise level. Therefore, the more difference between two views, the bigger ϵ should be defined. This is because gaits from very different views contain less common motion information (higher noise level). The SVR model should be more flexible than the model for closer views. Cherkassky et al. [124] proposed the following empirical dependency.

$$\epsilon = \tau\sigma\sqrt{\frac{ln\ N_s}{N_s}} \tag{4.19}$$

where τ is the constant, σ is the standard deviation of noise, and N_s is the number of training data samples.

The second main parameter is the variable C in Equation (2.10), the trade-off between the model complexity and the degree to which deviations larger than ϵ are tolerated in the optimization formulation. If C is too large, then the objective is to minimize the empirical risk only, regardless of the model complexity which might lead to model over-fitting. Cherkassky [124] proposed the use of the following prescription for the regularization parameter.

$$C = max(|\overline{y} + 3\sigma_y|, |\overline{y} - 3\sigma_y|) \tag{4.20}$$

where $\overline{y}$ is the mean of the training responses, and σ_y is the standard deviation of the training responses.

Besides, the parameters for each individual kernel should be properly determined because they also are the factors to control the performance of *SVR* process. These include d in Equation (2.14), the degree of the polynomial kernel and σ in Equation (2.15), the width of the Gaussian kernel. In this study, the use of cross-validation on the validation dataset is applied to estimate values of the kernel parameters. However, it could be automatically estimated in [125][126]. Once the initial values of required parameters is obtained, the optimal parameter set is found simultaneously by minimizing the prediction error on a selected validation dataset.

In this study, the implementation of *SVR* is based on the well known SVM-Light Support Vector Machine library [127].

4.6.2.3 Sparse Regression (SR)

In order to achieve the reliable regression for *rVTM*, the feature selection is essential to filter pixels in source *GEI*. Without a reliable feature selection, the regression process will definitely fail. By using *SR* as a regression tool, two layers of feature selection are applied.

The first layer is *ROI* selection as explained in Section 4.6.1. It is applied to ensure that selected *ROI* is highly correlated to the corresponding target pixel. *ROI* selection is a primary process of the feature selection to remove irrelevant information from source *GEI*. It is done based on the rough body segmentation and correlation strength. However, such rough *ROI* still contains redundant information because each pixel in *ROI* is selected independently (see Equation (4.15)). Pixels in *ROI* can be selected from overlapped motion areas, which provide the same information to the regression process. In order to prevent overfitting which will cause an unstable regression process, this work adds one more processing layer on top of the previous rough *ROI* selection. It will refine the selected *ROI* using the implicit feature selection functions embedded inside *SR*.

SR by the elastic net from Equation (2.7) is applied to train regression function (f_n) in Equation (4.11) and simultaneously reduce redundant information from selected *ROI*. This refined *ROI* can successfully avoid overfitting of regression model for $rVTM$ construction. The elastic net is applied to estimate a weight vector ($\hat{W}_n$) in Equation (4.11) as follow.

$$\begin{aligned}\hat{W}_n = (1+\lambda_2) \underset{W_n}{\operatorname{argmin}} \ ||g^n_{\theta_k} - \sum_{p=1}^{P} ROI^n_{\theta_i}(p) w_n(p)||^2_2 \\ + \lambda_2 \sum_{p=1}^{P} ||w_n(p)||^2_2 + \lambda_1 \sum_{p=1}^{P} ||w_n(p)||_1\end{aligned} \tag{4.21}$$

where λ_1 and λ_2 are non-negative values, $ROI^n_{\theta_i}(p)$ is the p^{th} pixel in $ROI^n_{\theta_i}$, $w_n(p)$ is the p^{th} index of weight vector W_n, and P is the number of pixels in *ROI*. The elastic net can be efficiently solved based on LARS algorithm [83]. As mentioned, the ridge penalty (L_2) is used to shrink coefficients of correlated predictors towards each other. Meanwhile, the lasso penalty (L_1) tends to select only one from highly correlated predictors. In Equation (4.21), L_1 and L_2 are combined to select proper predictors from selected *ROI* for the final regression model.

The refined *ROI* is ensured to have sufficient predictors with satisfying low redundancy to obtain stable and non-overfitting regression models for $rVTM$ construction in our proposed method. Figure 4.6 shows the refined *ROI* through *sparse regression* process. The red pixels in (d) are target pixels under target views. (a), (b) and (c) show different *ROI*s under source views (corresponding to each target pixel in (d)) produced by the different methods. The first two rows demonstrate two samples of *ROI*s produced for $rVTM_{0^\circ \to 18^\circ}$. The last two rows demonstrate two samples of *ROI*s produced for $rVTM_{90^\circ \to 108^\circ}$.

From Figure 4.6(a) and (d), it can be seen that the *ROI*s contain some pixels from arm region, while the corresponding target pixel are located in leg region.

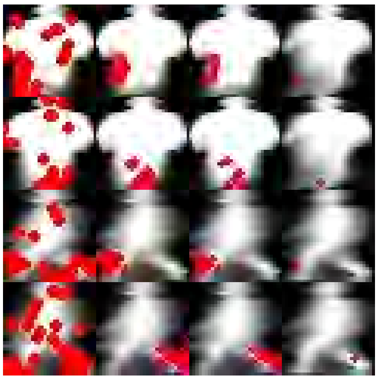


Figure 4.6: Examples of *ROI*s. The red areas in (a), (b) and (c) are *ROI*s produced by the different methods, which correspond to each red pixel in (d). The red pixels in (d) are the target pixels under the target view 18° (first 2 rows) and target view 108° (last 2 rows). (a) *ROI* produced by *SR* on the whole *GEI*. (b) *ROI* produced based on Equation (4.15) where T is 0.8. (c) Refined *ROI* using *SR* on the rough *ROI* regions shown in (b).

Based on our observation, these weak and non-generic relationships can lead to overfitting of regression models in *rVTM* construction. Thus, the rough body segmentation for computing *ROI* is an important component to maintain the reliability and performance of *rVTM*. From Figure 4.6(b) and (c), it can be seen that *SR* can eliminate some redundant pixels in the *ROI*s. This process also significantly helps to avoid overfitting of *rVTM*.

4.7 Gait similarity measurement

To measure the similarity between any two gait features (i.e. *GEI*s) from two different views $(G_{1,\theta_i}, G_{2,\theta_j})$, the gait features are first normalized to the same view (θ_k) by using the trained *VTM*(s) (either *dVTM* or *rVTM*) as follow.

$$\begin{aligned} G_{1,\theta_k} &= G_{1,\theta_i} \odot VTM_{\theta_i \rightarrow \theta_k} \\ G_{2,\theta_k} &= G_{2,\theta_j} \odot VTM_{\theta_j \rightarrow \theta_k} \end{aligned} \tag{4.22}$$

where $\theta_i \leq \theta_k \leq \theta_j$ given $\theta_i \leq \theta_j$. Next, the view-normalized gait features are projected into PCA subspace [68] using the PCA projection matrix $W_m \in \Re^{m \times N}$ (see Section 2.3.1) as follow.

$$\begin{aligned} E_1 &= W_m^T G_{1,\theta_k} = \{e_1^n\}_{n=1}^m \\ E_2 &= W_m^T G_{2,\theta_k} = \{e_2^n\}_{n=1}^m \end{aligned} \tag{4.23}$$

where W_m is calculated based on the training gait samples under view θ_k. Then, the simple but widely adopted Euclidean distance [119] is used to measure the gait dissimilarity (d) as follow.

$$d(G_{1,\theta_i}, G_{2,\theta_j}) = \sqrt{\sum_{n=1}^{m} (e_1^n - e_2^n)^2} \tag{4.24}$$

where the smaller value of d, the more possibility that the two gaits belong to the same subject.

4.8 Typical Implementations for Different Cases of Gait Recognition under Various Views

This section explains how to apply the VTM concept for different cases of gait recognition under various views. The performance of each case will be discussed in Section 4.9.

4.8.1 Cross-view gait recognition based on a one-view to one-view VTM

When a difference between two views in *cross-view gait recognition* (e.g. θ_i and θ_j) is small, then the view normalization requires only one VTM to either transform gaits from θ_i to θ_j or from θ_j to θ_i. For example, $\theta_i = 72°$ and $\theta_j = 90°$, then either $VTM_{72° \to 90°}$ or $VTM_{90° \to 72°}$ is required. The reason is mainly because VTM is very efficient and reliable for the transformation between close views.

4.8.2 Cross-view gait recognition based on one-view to one-view VTMs through middle view

In our study, it is observed that VTM is not very reliable for the transformation between far views. To improve the performance, two VTMs are required for the view normalization. That is, two VTMs are required to transform gaits to θ_k from θ_i and θ_j respectively. For example, $\theta_i = 72°$ and $\theta_j = 108°$. Instead of using single $VTM_{72° \to 108°}$ or $VTM_{108° \to 72°}$, $VTM_{72° \to 90°}$ and $VTM_{108° \to 90°}$ are used. As being explained in the first case, VTMs across two near views are more stable. This is because gaits from the smaller view difference share more common feature information.

4.8.3 Multi-view gait recognition based on a multi-view to one-view VTM

In practice, one-view to one-view transformation is not precise enough especially for the case of large view difference. This is because the orthogonality is degenerated when processing gait silhouette images. To overcome this problem, multi-view to one-view transformation is extended for *multi-view gait recognition* where gaits from multiple views are used to recognize gait from a single view. For example, gaits are recorded in the gallery dataset from N_v views, namely $\theta_1, \theta_2, ..., \theta_{N_v}$. They can be used to recognize probe gait on view θ_k by using multi-view to one-view VTM.

A multi-view to one-view $dVTM$ (which is extended from Equation (4.8)) is constructed as follow.

$$G_{s,\theta_k} = P_{\theta_k} \begin{bmatrix} P_{\theta_1} \\ . \\ . \\ . \\ P_{\theta_{N_v}} \end{bmatrix}^+ \begin{bmatrix} G_{s,\theta_1} \\ . \\ . \\ . \\ G_{s,\theta_{N_v}} \end{bmatrix} \tag{4.25}$$

On the other hand, each regression in a multi-view to one-view $rVTM$ (which is extended from Equation (4.11)) is modelled as follow.

$$\hat{g}^n_{\theta_k} = f_n(\hat{W}_n, ROI^n_{\theta_1} : ROI^n_{\theta_2} : ... : ROI^n_{\theta_{N_v}}) \tag{4.26}$$

where ":" is the concatenation between ROI vectors.

In our study, it has been seen that gaits from multiple views provide more information. Thus, Equations (4.25) and (4.26) will generate more precise view transformation results.

Multi-view to one-view VTM can be claimed as an implicit 3D gait model. The construction of VTMs is equivalent to the 3D model reconstruction and the view transformation using VTM is equivalent to the 3D rendering. However, the proposed VTM-based method might be more preferred because of the following reasons. First, it does not require expensive and complicated setup and maintenance of 3D system, camera calibration, and frame synchronization. Second, it contains a low computational complexity which is suitable for real-time applications.

4.8.4 Optimal view selection for multi-view gait recognition

In theory, the more number of cameras, the better performance can be obtained. However, in many practical situations, it is impossible to install too many cameras in a single location. Therefore, the practically workable idea is to choose a reasonable number of views which can construct VTMs to efficiently cover multi-view to one-view transformations.

Supposing there are N_v positions where a camera can be installed, due to certain practical constraints, we can install N_c cameras at N_c positions only ($N_c << N_v$). Therefore, there are totally C_{N_v,N_c} possible combinations ($C_{N_v,N_c} = \frac{N_v!}{N_c!(N_v-N_c)!}$ based on the permutation rule) to allocate N_c cameras from N_v positions. In [51], all possible combinations are testified for a given training dataset. The optimal combination is determined when the best gait recognition performance is achieved based on the training dataset. Depending on values of N_v and N_c, such kind of testing may take very long time.

In this study, a simple but efficient technique is introduced. It selects one view from each group of similar perspective views. The combination is then carried out from such representative views rather than from all views. A practical method is to categorize views into N_c groups first using any clustering method (e.g. k-mean clustering [128]) on gait features (i.e. *GEI*s). Then, the performance testing mentioned above is carried out for all combinations which select one view from each group. For example, $N_v = 11$ views are clustered into $N_c = 4$ groups containing 2,3,3,3 views. Then, total $2 \times 3 \times 3 \times 3 = 54$ experiments must be carried out. Otherwise, we need to execute $C_{11,4} = 330$ testing experiments as mentioned in [51].

4.9 Experiments

The CASIA gait database B is used in our experiments. The database is randomly divided into two groups. The first group of 24 subjects are used for: 1) construction of *VTM*; 2) learning of view classifier; 3) selection of optimal set of views. Typically, a larger number of training samples may contribute to a more efficient and generic *VTM* by learning from possible more variety gait samples. In our experiments, 24 subjects are used for the training processes. According to our observation, 24 subjects are enough to achieve satisfying training performance.

Then, the rest of 100 subjects in the second group are used for evaluating performance of: 1) automatic *view classification*; 2) gait recognition under various views. That is, *VTM* construction process is independent from subject recognition process. In this way, the generality of *VTM* is demonstrated. As an example, Figure 4.7 illustrates the transformed *GEI*s through the proposed *rVTM*-based method. It shows that the proposed method can effectively perform view transformation based on *GEI*s.

The experiments are completed using the computer machine with Quad Processor 2.66 GHz and 4 GB Ram. The training time for one *VTM* using the proposed method is approximately 10-20 minutes. However, in the testing phase, it takes only less than 0.001 second to achieve one view-transformation. Moreover, the training processes can be completed offline beforehand.

To compute each experimental result using the proposed method for *cross-view gait recognition* based on a one-view to one-view *VTM*, probe gait data is transformed to a feature set under view that matches one of the views in the gallery gait dataset. Then, gait similarity is measured using Euclidean distance in *PCA* subspace and Nearest Neighbor (NN) is used as a classification method.

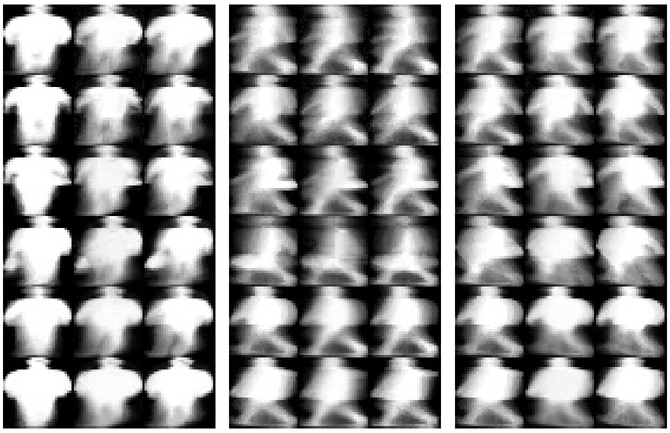


Figure 4.7: Examples of transformed *GEI*s. Each row is obtained from a different subject. The first two rows are examples of normal walking, the third and fourth rows are examples of walking when carrying a bag, and the last two rows are examples of walking when wearing a coat. (a) is G_{0°. (b) is transformed G_{18° from (a). (c) is corresponding G_{18°. (d) is G_{126°. (e) is transformed G_{108° from (d). (f) is corresponding G_{108°. (g) is G_{126°. (h) is transformed G_{144° from (g). (i) is corresponding G_{144°.

4.9.1 View classification

The proposed *view classification* based on *LDA* in *PCA* transformed space (see Section 4.3) is compared with the methods in [48] using GP- or SVM-based classification. It can be seen from Table 4.1 that the proposed classifier gives better overall results than SVM- and GP-based classifiers.

To further evaluate the performance of the proposed *view classification*, we compare overall performance of *cross-view gait recognition* where view is automatically classified with the case where view is given manually beforehand. By follow the case explained in Section 4.8.1, the performance of *cross-view gait recognition* is tested. Table 4.2 shows the evaluation results. In this evaluation, it is essential to build up $VTM_{probe \rightarrow gallery}$. View of probe data can be automatically classified using the proposed method (1^{st} case mentioned above) or can be given manually beforehand (2^{nd} case mentioned above). The first two rows of Table 4.2 list views of probe data and gallery data respectively. $A\%$ in the table stands for accuracy of *view classification* (i.e. view of probe data being classified

Table 4.1: Rank 1 *view classification* results (%) for each view from the CASIA gait database B.

Method	0°	18°	36°	54°	72°	90°	108°	126°	144°	162°	180°	Average
SVM [48]	–	–	95	41	85	64	24	44	98	–	–	64
GP [48]	–	–	84	91	85	74	86	91	94	–	–	86
Our method	90	91	91	98	99	91	93	94	92	93	91	94

Table 4.2: Influence of automatic *view classification* on the overall performance of cross-view gait recognition.

	Probe	0°	0°	54°	54°	90°	90°	126°	126°
	Gallery	180°	18°	36°	72°	72°	108°	108°	144°
Performance of *view classification* and the relevant cross-view gait recognition (A%,B%)	0°	(90,88)	(90,78)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
	18°	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
	36°	(0,0)	(0,0)	(2,0)	(2,1)	(0,0)	(0,0)	(0,0)	(0,0)
	54°	(0,0)	(0,0)	(98,95)	(98,93)	(0,0)	(0,0)	(0,0)	(0,0)
	72°	(0,0)	(0,0)	(0,0)	(0,0)	(2,2)	(2,1)	(0,0)	(0,0)
	90°	(0,0)	(0,0)	(0,0)	(0,0)	(91,87)	(91,88)	(0,0)	(0,0)
	108°	(0,0)	(0,0)	(0,0)	(0,0)	(7,5)	(7,3)	(5,3)	(5,3)
	126°	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(94,91)	(94,90)
	144°	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(1,0)	(1,0)
	162°	(1,1)	(1,1)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
	180°	(9,6)	(9,6)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
Result1		95	85	95	94	94	92	94	93
Result2		98	88	99	98	97	96	99	98

as the corresponding view listed in the first column). $B\%$ in the table stands for accuracy of *cross-view gait recognition* after view of probe data being classified as the corresponding view listed in the first column. The last two rows in the table are overall performance of *cross-view gait recognition*.

'Result1' is the overall performance when automatic *view classification* is adopted, which in fact equals to the sum of $B\%$ in each experiment (i.e. each column). 'Result2' is the performance when view of probe data is known in any way beforehand (e.g. provided by the CASIA gait database B). From the results, it is seen that overall *cross-view gait recognition* with the proposed automatic *view classification* is comparable to the case where view is precisely known be-

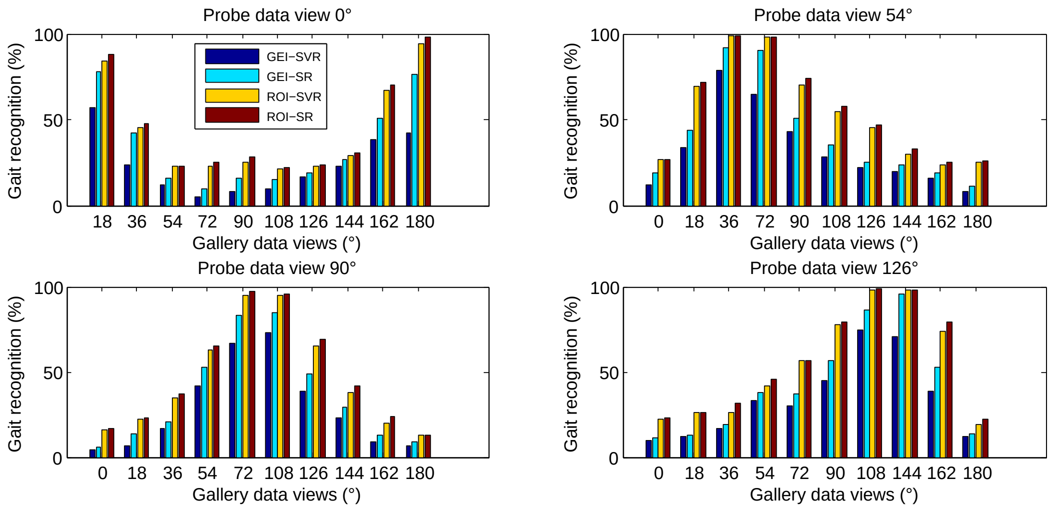


Figure 4.8: Comparisons on cross-view gait recognition (%) using the proposed *rVTM*-based methods without *ROI* selection (*GEI-SVR*, *GEI-SR*) and the proposed *rVTM*-based methods with *ROI* slelection (*ROI-SVR*, *ROI-SR*).

forehand. However, it is obvious that introducing automatic *view classification* can improve usability of *cross-view gait recognition* in the real world.

4.9.2 Performance evaluation on ROI selection

This section is used to evaluate the performance on *ROI* selection in our *rVTM* construction as mentioned in Section 4.6.1. *Cross-view gait recognition* based on a one-view to one-view *VTM* (see Section 4.8.1) is evaluated using two versions of the framework including: 1) the framework without *ROI* selection (i.e. *GEI* is used as input for each regression in *rVTM* construction); 2) the framework with *ROI* selection.

Two regression techniques (i.e. *SVR* and *SR*) are used as a case study. Accordingly, the experiments are carried out using four *rVTM*-based methods: 1) *GEI-SVR*; 2) *GEI-SR*; 3) *ROI-SVR*; 4) *ROI-SR*. *GEI-SVR* and *GEI-SR* are the proposed *rVTM*-based methods without *ROI* selection by using *SVR* and *SR* respectively as a regression tool. *ROI-SVR* and *ROI-SR* are the proposed *rVTM*-based methods with *ROI* selection by using *SVR* and *SR* respectively as a regression tool. RBF kernel is used for *SVR*.

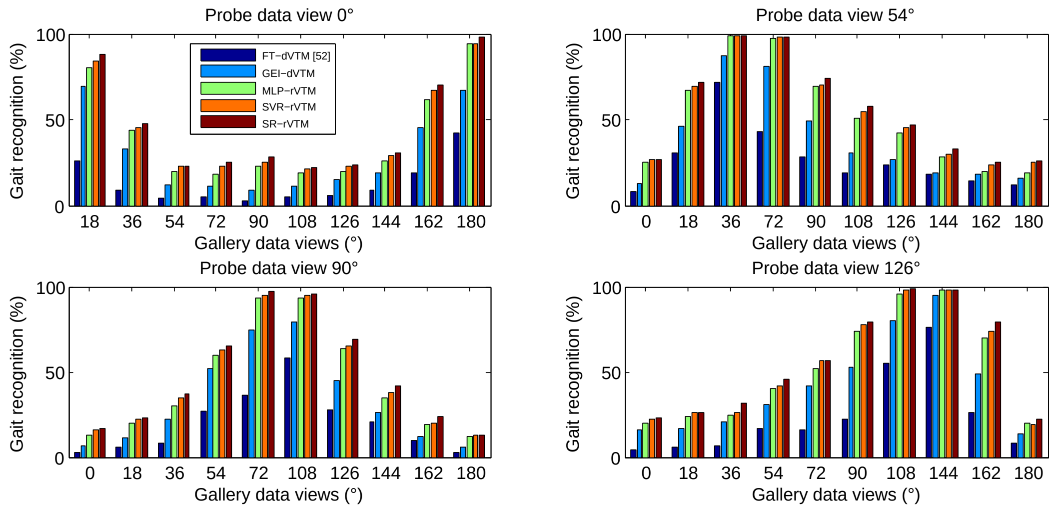


Figure 4.9: Comparisons on cross-view gait recognition (%) using different *VTM*-based methods including FT-*dVTM*, *GEI-dVTM*, *MLP-rVTM*, *SVR-rVTM*, and *SR-rVTM*.

From Figure 4.8, it can be seen that without *ROI* selection, the performances of *GEI-SVR* and *GEI-SR* significantly decrease. However, *GEI-SR* still achieves much better performance than *GEI-SVR* due to the implicit feature selection property inside *SR*. It proves the good feature selection performance embedded in *SR* and also sparsity of *GEI* in regression process.

From Figure 4.8, with *ROI* selection, *ROI-SVR* and *ROI-SR* can achieve significantly better performance. As mentioned, this is because *ROI* selection can remove noises and uncorrelated information from *GEI* for each regression process in *rVTM* construction.

Another important benefit of *ROI* selection is to reduce computational complexity. In this study, *GEI* contains 900 pixels while *ROI* contains approximately 30 pixels. That is, *ROI* selection process can reduce dimension of regression's input from 900 to 30. Average time to train one *rVTM* using *ROI* as regression's input is approximately 10 minutes, while it takes approximately 6 hours to train one *rVTM* using *GEI* as regression's input.

ROI selection process is taken in the rest of experiments to further evaluate the performance of the proposed *rVTM*-based method.

4.9.3 Cross-view gait recognition based on a one-view to one-view VTM

The performance of *cross-view gait recognition* proposed in Section 4.8.1 is evaluated. First, the experiments are carried out to verify efficiency of *VTM*-based methods. Accordingly, the experiments are carried out using five methods: 1) FT-*dVTM*; 2) *GEI-dVTM*; 3) *MLP-rVTM*; 4) *SVR-rVTM*; 5) *SR-rVTM*. FT-*dVTM* [52] is the conventional method to construct *dVTM* based on the frequency-domain gait feature using Fourier Transfer (FT). *GEI-dVTM* is the proposed *dVTM*-based method by using T*SVD* on the optimized *GEI*. *MLP-rVTM*, *SVR-rVTM*, and *SR-rVTM* are the proposed *rVTM*-based methods by using *MLP*, *SVR*, and *SR* respectively as a regression tool.

Based on our observations (see our published work in [49] for the detailed experimental results), among the three kernels for *SVR*, it can be seen that RBF kernel provides the highest accuracy, followed by polynomial kernel and linear kernel respectively. Thus, in this section, only the results of RBF kernel is shown for *SVR-rVTM*.

From Figure 4.9, several key points are concluded as below.

- Among *dVTM*-based methods, the proposed *GEI-dVTM* significantly outperforms FT-*dVTM* [52]. This is because of the following reasons. First, *GEI* in the spatial-domain can achieve more efficiency than FT-based gait feature in the frequency-domain. *GEI* contains more non-zero or meaningful pixels than FT-based gait feature, which can be considered as a better feature for *SVD* matrix decomposition process. Second, *LDA* acquires optimized *GEI* in the proposed method. This improves the performance of *dVTM* construction. In the experiments, *LDA* significantly reduces dimension on the optimized *GEI* by 90% of the original *GEI*. It captures projected principal gait features with 99% significance. Third, compare to *SVD* used in FT-*dVTM* [52], T*SVD* used in the proposed method can improve *dVTM* construction by avoiding overfitting. T*SVD* is strongly preferable for online real-time applications. This is because T*SVD* can significantly reduce the size of *dVTM* by 40% from the original size using *SVD*. Thus, it can also achieve the reduction of computational time and memory storage.

- Among *rVTM*-based methods, it can be seen that *SR-rVTM* provides the highest accuracy, followed by *SVR-rVTM* and *MLP-rVTM* respectively. As mentioned in Section 4.6.2.3, it is because of additional feature selection embedded inside *SR*. This benefit can not be obtained by adopting *MLP* or *SVR*. Moreover, *SVR-rVTM* can perform better than *MLP-rVTM* because of the sparseness of *SVR* on noisy training dataset and its potential power of non-linear kernel (i.e. RBF).

- Particularly, *rVTM* (i.e. *MLP-rVTM*, *SVR-rVTM*, *SR-rVTM*) significantly outperforms *dVTM* (i.e. FT-*dVTM* [52], *GEI-dVTM*) in all cases. *SR-rVTM* achieves the accuracy up to 95% in average for the case of small view-difference ($\pm 18°$). However, FT-*dVTM* [52] and *GEI-dVTM* achieves the accuracy only up to 70% and 85% respectively. This is because *rVTM* contains more solid and realistic problem formulation. Moreover, it is shown that selecting local features for the regression processes in *rVTM* instead of global features as being used in *dVTM* provides better performance for gait recognition under view changes. The local features are more robust for the view transformation process.

- In addition, it is obviously seen that transformation of gaits between two closer views results in the better performance because the gaits share more common motion information.

Since *SR-rVTM* achieves the best performance among the proposed *VTM*-based methods as discussed above, it will be used for the rest of the experiments in this chapter.

Next, comprehensive comparisons are carried out between the proposed *SR-rVTM* and other non *VTM*-based methods that also directly address the problem of view change for gait recognition. Figure 4.10 illustrates the first rank for *cross-view gait recognition* by using four different methods: 1) the baseline method which simply matches *GEI* across views without any view transformation [3]; 2) view rectification using self-calibrating [44]; 3) GFI-*CCA* [48]; 4) *SR-rVTM* (the proposed method). Some missing results have not been reported by [44] and [48].

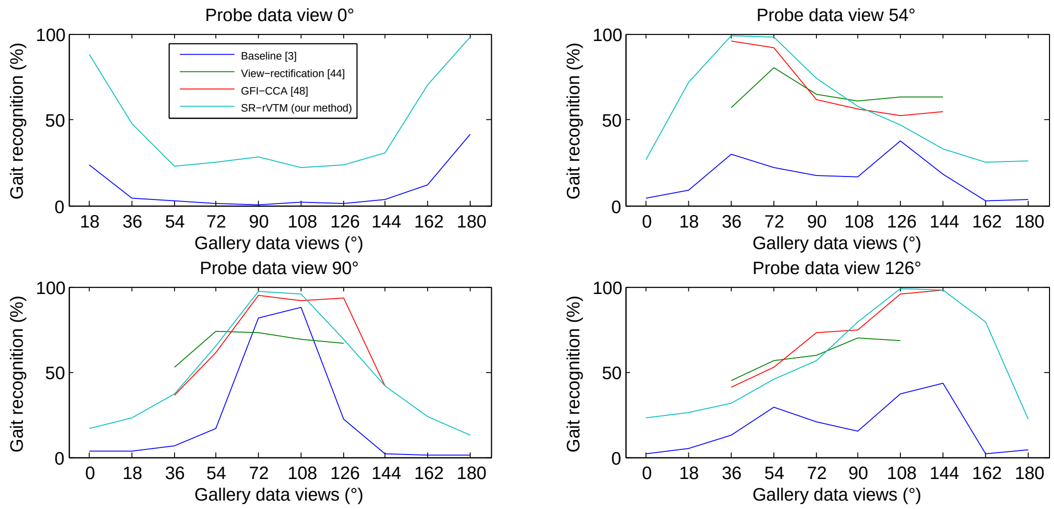


Figure 4.10: Comparisons on cross-view gait recognition based on an one-view to one-view *VTM* (%) using different methods.

From Figure 4.10, it can be seen that the proposed method significantly outperforms the baseline method [3]. It also performs better than [44][48] for the cases of small view change, but worse for some cases of large view change. In our method, this limitation of large view change can be overcome using: 1) one-view to one-view *VTM*s through middle view (see Section 4.8.2 and 4.9.4); 2) a multi-view to one-view *VTM* (see Section 4.8.3 and 4.9.5) for *multi-view gait recognition*, which have not been discussed by [44][48].

4.9.4 Cross-view gait recognition by transforming gaits under source and target views into another common view

As mentioned, when a difference between views of probe and gallery gaits is relatively large, *cross-view gait recognition* by transforming one view directly into another view will not perform well. A practical solution is to transform both views into a common view in the middle (see Section 4.8.2). Then, gait similarity is measured on this new view. This is because it is more reliable to transform gait feature to a closer view.

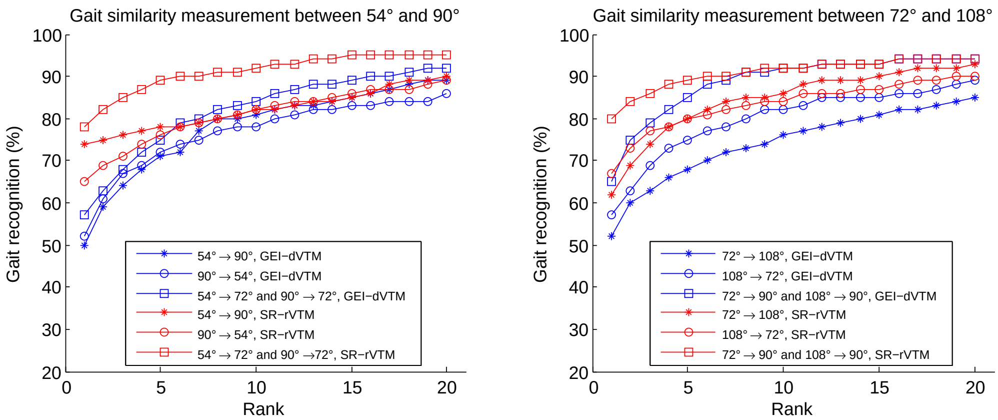


Figure 4.11: Comparisons on *cross-view gait recognition* based on one-view to one-view VTMs through middle view (%) using different methods.

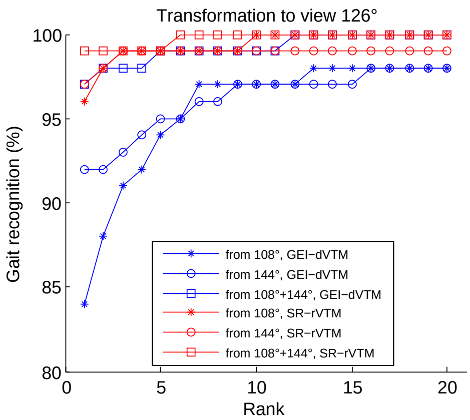
Cumulative Match Score (CMS) curves are drawn in Figure 4.11 to verify this solution. CMS with rank r means it indicates fraction of probes yielded a correct match within the top r candidates. It also can be seen that $rVTM$ outperforms $dVTM$ under this same scenario.

4.9.5 Multi-view gait recognition based on a multi-view to one-view VTM

This section is used to evaluate the performance of multi-view to one-view VTM for *multi-view gait recognition* where gaits from multiple views are used to recognize gait from a single view (see Section 4.8.3). *Multi-view gait recognition* using the proposed VTM-based method does not require any multi-camera system, camera calibration, and video synchronization.

To generate results in Figure 4.12, gait feature(s) from gallery dataset is/are transformed to match view of probe gait feature. Figure 4.12 shows that the performance of *multi-view gait recognition* using two-view to one-view transformation is significantly better than *cross-view gait recognition* using standard one-view to one-view transformation.

The reason is simply because gaits from multiple views provide more sufficient information for the view transformation process. It also can be seen that



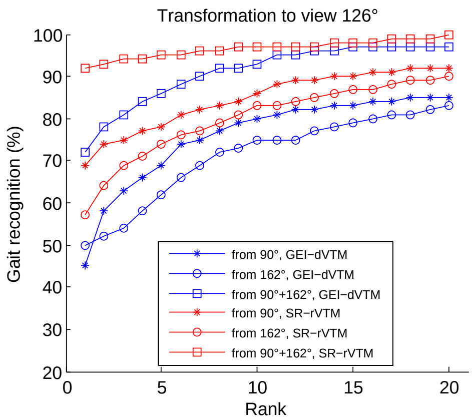


Figure 4.12: Comparisons on *multi-view gait recognition* based on a multi-view to one-view *VTM* (%) using different methods.

Table 4.3: Evaluation of *multi-view gait recognition* (%) using the proposed method.

Number of gallery views	0°	18°	36°	54°	72°	90°	108°	126°	144°	162°	180°
3	100	92	81	91	99	96	85	97	98	97	97
4	100	99	99	100	96	97	99	98	96	100	98

rVTM (e.g. *SR-rVTM*) outperforms *dVTM* (e.g. *GEI-dVTM*) for *multi-view gait recognition*.

4.9.6 View selection for multi-view gait recognition

This section demonstrates the performance of our view selection for *multi-view gait recognition*. The CASIA gait database B which is employed in this experiment, contains 11 views. Two systems are tested by using gaits under 3 and 4 gallery views respectively to recognize gaits under any 11 views.

Based on Section 4.8.4, the 11 views are clustered into groups using k-mean clustering [128] on *GEI*s. For the system with 3 gallery views, the 11 views are clustered into 3 groups as group1: $0°, 18°, 162°, 180°$, group2: $36°, 54°, 126°, 144°$ and group3: $72°, 90°, 108°$. For the system with 4 gallery views, the 11 views

Table 4.4: Comparisons on *multi-view gait recognition* (%) using different methods.

Method	REI+*LDA* [51] using 3 gallery views	The proposed method using 3 gallery views
Average performance	91	94
Method	3D model-based method [41] using 4 gallery views	The proposed method using 4 gallery views
Average performance	70	98

are clustered into 4 groups as group1: $0°, 180°$, group2: $18°, 36°, 162°$, group3: $54°, 126°, 144°$ and group4: $72°, 90°, 108°$.

Based on our evaluation on the training dataset, the optimal views for the systems with 3 and 4 gallery views are $\{0°, 72°, 144°\}$ and $\{0°, 54°, 108°, 162°\}$ respectively. To evaluate the recognition performances of the systems on the testing dataset, for example of the system with 4 gallery views, gaits from views 0°, 54°, 108° and 162° are used to train 11 *VTM*s. Then, $VTM_{0°+54°+108°+162°\to 0°}$, ..., $VTM_{0°+54°+108°+162°\to 180°}$ are used to recognize gait from each single view (see Section 4.8.3). The similar experiment is carried out for the system with 3 gallery views.

Refer to Table 4.3, the proposed method (i.e. *SR-rVTM*) achieves very high performance for recognizing gaits under each single view by using only 3 or 4 gallery views. Refer to Table 4.4, by using 3 gallery views, the proposed method achieves average accuracy of 94% which is better than the method in [51]. By using 4 gallery views, the proposed method significantly outperforms the 3D model-based method [41] which can achieve about 70% accuracy. This is because 3D gait motion is a complicated dynamic shape model. The 3D gait analysis is usually interfered by noises from various factors such as low resolution video, poor foreground segmentation, shadow, partial occlusion, etc. which directly affect the performance of its analysis.

Table 4.5: Performance of *cross-view gait recognition* (%) under various walking conditions.

Probe	Gallery	Separated *VTM*			Mixed *VTM*		
		Normal	With a bag	With a coat	Normal	With a bag	With a coat
0°	180°	98	94	95	96	93	94
0°	18°	88	85	85	82	81	83
54°	36°	99	94	95	95	92	94
54°	72°	98	89	94	94	88	91
90°	72°	97	87	93	93	86	90
90°	108°	96	89	92	93	87	90
126°	108°	99	93	97	94	92	96
126°	144°	98	94	95	94	91	94

4.9.7 Cross-view gait recognition under various conditions

In this experiment, we consider 3 different walking conditions including: 1) normal walking (without carrying a bag and without wearing a coat); 2) walking when carrying a bag; 3) walking when wearing a coat. Gait recognition is carried out on each walking condition separately. However, *VTM* (i.e. *SR-rVTM*) is trained using two different ways including: 1) *VTM* is trained for each walking condition separately; 2) *VTM* is trained for mixed walking conditions. This is used to verify robustness of *VTM* under mixed walking conditions when compared with *VTM* under single walking condition.

From Table 4.5, *VTM* from mixed walking conditions gives just slightly worse performance than *VTM* from single walking conditions. That is, when *VTM* is constructed using the training dataset which covers sufficient walking samples, *VTM* can be efficiently used for the view transformation of gaits under various walking conditions.

4.9.8 Cross-view gait recognition under outdoor environment

The proposed method is further evaluated using the practical dataset (outdoor) i.e. the USF gait database. This database is challenging because of several

Table 4.6: Performance of *cross-view gait recognition* (%) under outdoor environment using the experiment A of the USF gait database.

[2]	[27]	[28]	[129]	[130]	[131]	[132]	Our method
73	83	85	83	80	86	80	89

difficulties of the outdoor environment such as wind, shadow, and illumination. The USF gait database contains a set of 12 challenge experiments which are designed to investigate the effect of five factors affecting the performance of gait recognition. Among the 12 experiments, the experiment A is adopted for our evaluation because it observes view change for gait recognition which matches the focus of this study.

In the experiment A, probe and gallery gaits are recorded from different cameras L and R respectively. In our experiments, 22 subjects are used for *VTM* (i.e. *SR-rVTM*) construction. The rest of 100 subjects are used for evaluating *cross-view gait recognition*. From Table 4.6, the proposed method (which achieves 89%) outperforms other methods in the literature (which achieve below 86%).

4.10 Discussion and Conclusion

This chapter has proposed the methods for gait recognition under view change based on two types of *VTM*, *dVTM* and *rVTM*. In this study, *dVTM* is constructed based on the matrix factorization process by using T*SVD* on the optimized *GEI*. T*SVD* is applied to factorize the gait matrix on the training dataset into the subject-independent matrix and the view-independent matrix. Such subject-independent matrix is dependent to the trained views and then is used to build up *dVTM*. Thus, in the testing phase, *dVTM* can be used for the view transformation process in generic for any seen/unseen subject. Based on our experiments, the proposed *dVTM*-based method is shown to significantly outperform the conventional *dVTM*-based method, by applying the reduced version of *SVD* and the optimized version of *GEI*. However, its performance is still limited due to its linear processes on global features.

In this thesis, *VTM* is reformulated as the newly proposed *rVTM*. *rVTM* is constructed based on the concept of correlated motion regression. Regression processes are used to formulate and model correlated motions among gaits across different views. To achieve a reliable regression, *ROI* selection is a core process to filter source gait feature and remain with only relevant information to predict corresponding information in target gait feature. In this study, three regression techniques including *MLP*, *SVR*, and *SR* are independently adopted as regression tools for *rVTM* construction. As a result, *SR* provides the highest performance. This is because *SR* can reduce redundancy in *ROI* based on its sparse properties. Such refined *ROI* can be used to generate more stable and non-overfitting regression fitted models in *rVTM* construction. Based on our experiments, *rVTM*-based methods are shown to be significantly better than *dVTM*-based methods in all cases. Consequently, the proposed *rVTM*-based method (i.e. *SR-rVTM*) also outperforms the other methods in the literature.

In fact, any regression tool can be applied for the *rVTM* construction by following the framework proposed in this study. The proposed method can be also applied to any suitable gait feature, if it is able to preserve correlation between motions from different views. This property is a fundamental assumption of *rVTM*. To further improve the efficiency of our *rVTM* construction, the process of *ROI* selection should be improved to be more reliable and robust for different realistic situations such as walking when carrying a bag and partial occlusions.

To improve the gait recognition performance, the proposed *VTM*-based method is further applied for *multi-view gait recognition* using multi-view to one-view transformation. The experiments show that multi-view to one-view transformation significantly outperforms standard one-view to one-view transformation in all cases. As mentioned in Section 4.8.3, the proposed multi-view to one-view *VTM* can be claimed as an implicit 3D model-based method. However, it is shown to be more efficient than the 3D gait model-based method for *multi-view gait recognition*. Moreover, considering real applications of gait recognition under various views, this chapter has also proposed an efficient *view classification* method and discusses several implementation matters under the different cases.

As mentioned in the introduction of this chapter, the methods in the third category [48][49][50][51][52] including the proposed *VTM*-based method cannot be used for the view-normalization of untrained views because they rely on supervised learning. However, these methods can be extended in somehow for *cross-view gait recognition* of any trained/untrained view using the following technique.

In fact, current gait feature can tolerate small change of view. That is, the gait feature of untrained view can be approximated by another one of nearby trained view as long as the gap between these two views is less than tolerable limitation mentioned above. For *cross-view gait recognition* based on *VTM*, we may use sufficient number of cameras to record gaits under multiple views which can be used to train a sufficient number of *VTM*s. The gap between any two consecutive views is smaller than the largest view-offset which can be tolerated by the adopted gait feature. So, *VTM*s which are constructed with such camera setup approximately, can cover all views. By using a more robust gait feature, a less number of cameras will be required for *VTM* construction process.

Chapter 5

Gait Recognition under Various Views based on Correlated Motion Co-clustering

5.1 Introduction

In recent years, several methods (see Section 1.2.2) have been proposed from different perspectives for gait recognition under view change. The existing state-of-the-art [48][49] relies on analysis of correlated walking motions across views. The efficient method using *rVTM* was introduced in [49] (i.e. our method proposed in Chapter 4). *rVTM* was used to transform correlated gait information from one view onto another view. Its regression framework is based on mapping between a group of pixels, which is called 'segment' in this chapter, and a single-pixel. Such mapping could become unstable when the pixels shift due to an occlusion and/or a shadow on gait. The *VTM*-based methods may also suffer from degeneracies and singularities caused by features visible in one view but not in others, especially in the case of large view changes.

The method in [48] is another recent published state-of-the-art procedure, which projected global gait features from two different views into two subspaces for achieving correlation maximization. This method regarded the global gait feature as a whole in the projection procedure. The performance is limited when the unrelated gait information from different views is involved in the projection.

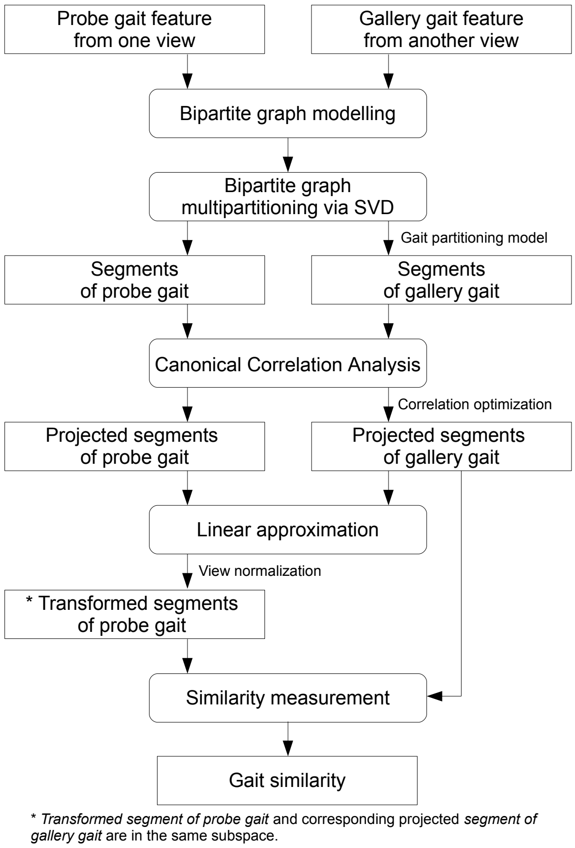


Figure 5.1: The proposed framework for *cross-view gait recognition*.

Obviously, when view changes, certain gait information may disappear from one view to the others. Such gait information will not contribute to the gait recognition under various views. Instead, it will be a barrier for recognizing gaits if it is not carefully eliminated.

This chapter proposes a novel method that can overcome the limitations of the current state-of-the-art methods [48][49]. The proposed method relaxes the mapping relationship across views by using a segment to segment mapping through a correlated motion co-clustering method. The correlation of gaits from different views is then maximized based on the corresponding local segments rather than on the global gait features. In this way, the correlation maximization of gaits from different views will be further optimized by involving only the related gait information in the local segments.

Figure 5.1 shows the framework of the proposed solution for *cross-view gait* *recognition*. Given a training dataset containing individual gaits from two different views, our framework contains three main steps in the training process, namely 1) *gait partitioning* model using *bipartite graph multipartitioning* (see Section 5.3); 2) correlation optimization using *CCA* (see Section 5.4); 3) view normalization using *linear approximation* (see Section 5.4).

The first step is to learn the *gait partitioning* model for *cross-view gait recognition*. A bipartite graph is used to model the correlation between gaits from two different views, then a *bipartite graph multipartitioning* is applied to co-cluster gaits across the two views into segments. Consequently, the corresponding segments are the most correlated and contain the most similar gait information. The second step is to maximize the correlation between gaits from different views. *CCA* is then applied to project the corresponding segments from the two views into two subspaces where their linear correlation is maximized. Such subspaces can be called *CCA* subspaces. The final step is to learn a *linear approximation* model in order to linearly transform the corresponding segments of gaits from the two *CCA* subspaces into the same *CCA* subspace.

In the testing phase, probe and gallery gaits are co-clustered into segments using the relevant (i.e. regarding their views) trained *gait partitioning* model. Then, the correlation optimization model (i.e. *CCA* projection matrices) and the *linear approximation* model which have been obtained in the training process, are applied to the gait segments to project them onto a common *CCA* subspace where the similarity measurement can be carried out properly.

When compared with the state-of-the-art [48][49], the proposed method demonstrates the following advantages.

- By co-clustering correlated motion patterns across views, the proposed method achieves a robust mapping between a gait segment under one view and the other most correlated gait segment under another view in the optimized subspaces.

- Since the proposed method is performed on the local gait segments rather than the global gait feature, it can avoid spurious correlations.

- When compared with other methods in the literature, the proposed method can enhance gait recognition under large view change significantly.

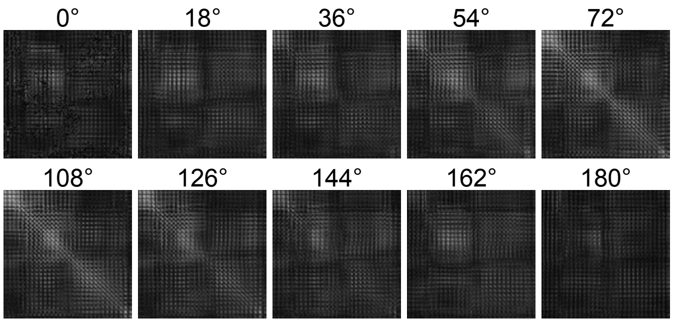


Figure 5.2: The correlations between *GEI*s from $90°$ and other views (i.e. $0°, 18°, 36°, 54°, 72°, 108°, 126°, 144°, 162°, 180°$) as indicated on the top of each image. The lighter color represents the higher correlation.

- The proposed method can be extended for *multi-view gait recognition* with very encouraging performance, where camera cooperation and geometry constraints are not required.

The rest of this chapter is organized as follows. Adopted gait feature is explained in Section 5.2. *Gait partitioning* model is proposed in Sections 5.3. Correlation optimization and *linear approximation* are described in Section 5.4. *Cross-view* and *multi-view gait recognitions* are discussed in Section 5.5. Experimental results are shown in Section 5.6 and conclusions are drawn in Section 5.7.

5.2 Gait Feature Preparation

GEI (see Section 2.1) is also used as a gait feature in this chapter. This is because it contains high correlations of walking motions across different views. The preprocessing including *gait period* estimation and *view classification* are explained in Sections 3.2 and 4.3.

As mentioned in Section 2.1, although *GEI*s under various views are different, they are captured from the same 3D gait of an individual. Therefore, they must share correlated information especially for the case of small view difference. This is analyzed as follows.

In Figure 5.2, each image shows the correlation between *GEI*s from $90°$ and another view as indicated on the top of the images. In each image, columns repre-

Table 5.1: Table of notations related to gait features.

Symbol	Explanation
$G_{\theta_g}, G_{\theta_p}$	GEIs from views θ_g and θ_p respectively
$G_{n,\theta_g}, G_{n,\theta_p}$	The n^{th} samples of G_{θ_g} and G_{θ_p} respectively
$G^k_{\theta_g}, G^k_{\theta_p}$	The k^{th} segments of G_{θ_g} and G_{θ_p} respectively
$p^i_{\theta_g}, p^i_{\theta_p}$	The i^{th} pixels of G_{θ_g} and G_{θ_p} respectively
$p^i_{n,\theta_g}, p^i_{n,\theta_p}$	Values of the i^{th} pixels of G_{n,θ_g} and G_{n,θ_p} respectively
N_b, N_d	Sizes (i.e. total numbers of pixels) of G_{θ_g} and G_{θ_p} respectively **Note:** N_b and N_d are not necessarily the same
N_g, N_p	Sizes (i.e. total numbers of pixels) of $G^k_{\theta_g}$ and $G^k_{\theta_p}$ respectively **Note:** N_g and N_p are not necessarily the same

sent pixels of GEI from 90° and rows represent pixels of GEI from another view. The correlation coefficient (see Equation (5.4)) between each pixel of GEI from 90° and each pixel of GEI from another view is calculated and then illustrated in the images of Figure 5.2. The lighter color shows the higher correlation. Twenty four subjects from the CASIA gait database B are used in this analysis.

From Figure 5.2, several important points can be observed. First, neighboring pixels are highly correlated because they likely belong to the same/correlated body parts. Second, GEIs from closer views share stronger correlations because they share more visual information. Third, not all pixels of GEI are correlated with each other. Thus, it will be more efficient to bridge gaps between GEIs from different views on the local correlated segments rather than normalization on the global GEI.

Moreover, to avoid any confusion, Table 5.1 lists key notations related to gait features which are frequently used in this chapter.

5.3 Gait Partitioning Model

As analyzed in Section 5.2, not all information of *GEI*s from different views is correlated. Thus, *GEI*s should be co-clustered into segments such that the corresponding segments across the views contain the most correlated gait information. Sequentially, the correlation optimization can be applied on the corresponding segments instead of the global *GEI*s, in order to avoid spurious relationships between unrelated information.

In this section, *gait partitioning* model is proposed to achieve this co-clustering using a *bipartite graph multipartitioning* (see Section 2.6). By describing gaits from different views (G_{θ_g} and G_{θ_p}) with a bipartite graph, partition such a graph can yield the co-clustering of G_{θ_g} and G_{θ_p}.

The *gait partitioning* model is learned from a training dataset, which is explained below. The training dataset is given as $\{G_{n,\theta_g}, G_{n,\theta_p}\}_{n=1}^{N_a}$ where each pair contains *GEI* samples of the same subject from different views, G_{n,θ_g} and G_{n,θ_p} are the n^{th} samples of G_{θ_g} and G_{θ_p} respectively, and N_a is the total number of training samples.

5.3.1 Problem formulation based on co-clustering

Given $G_{\theta_g} = \{p^i_{\theta_g}\}_{i=1}^{N_b}$ and $G_{\theta_p} = \{p^i_{\theta_p}\}_{i=1}^{N_d}$ where $p^i_{\theta_g}$ and $p^i_{\theta_p}$ are the i^{th} pixels in G_{θ_g} and G_{θ_p} respectively, and N_b and N_d are the total number of pixels in G_{θ_g} and G_{θ_p} respectively, G_{θ_g} and G_{θ_p} are simultaneously clustered into N_c disjoint segments as follow.

$$\begin{aligned} G_{\theta_g} &= G^1_{\theta_g} \cup ... \cup G^{N_c}_{\theta_g} \\ G_{\theta_p} &= G^1_{\theta_p} \cup ... \cup G^{N_c}_{\theta_p} \end{aligned} \tag{5.1}$$

where the corresponding segments ($G^k_{\theta_g}$ and $G^k_{\theta_p}$, $1 \leq k \leq N_c$) are the most correlated.

That is, a given pixel $p^i_{\theta_g}$ belongs to $G^k_{\theta_g}$ if its correlation with $G^k_{\theta_p}$ is greater than its correlation with any other segments in G_{θ_p}. Thus, given $G_{\theta_p} = G^1_{\theta_p} \cup ... \cup G^{N_c}_{\theta_p}$, the corresponding segmentation of G_{θ_g} will be determined as follow.

$$G^k_{\theta_g} = \{p^i_{\theta_g} : \sum_{p \in G^k_{\theta_p}} \gamma(p^i_{\theta_g}, p) \geq \sum_{p \in G^m_{\theta_p}} \gamma(p^i_{\theta_g}, p),\ \forall m = 1, ..., N_c\} \tag{5.2}$$

where γ is an absolute value of the correlation coefficient between the two pixel-variables [2]. In the same way, when given $G_{\theta_g} = G^1_{\theta_g} \cup ... \cup G^{N_c}_{\theta_g}$, the induced segmentation of G_{θ_p} is given by follow.

$$G^k_{\theta_p} = \{p^i_{\theta_p} : \sum_{p \in G^k_{\theta_g}} \gamma(p^i_{\theta_p}, p) \geq \sum_{p \in G^m_{\theta_g}} \gamma(p^i_{\theta_p}, p),\ \forall m = 1, ..., N_c\} \tag{5.3}$$

From Equations (5.2) and (5.3), it can be seen that co-clustering of *GEI*s from different views is recursive in nature. This is because segments of G_{θ_g} determine segments of G_{θ_p} which in turn determine segments of G_{θ_g}. This process can be achieved by a *bipartite graph multipartitioning*. In general, the proposed *gait partitioning* method can be applied to other types of gait features, which is not limited to *GEI*.

5.3.2 Bipartite graph modelling on gait features from different views

The bipartite graph is used in this study to model correlations between gaits from different views. The bipartite graph is a triple $B = (V, E, W)$ where $V = G_{\theta_g} \cup G_{\theta_p}$ contains two sets of vertices representing pixels in *GEI*s, $E = \{(p^i_{\theta_g}, p^j_{\theta_p}) : p^i_{\theta_g} \in G_{\theta_g}, p^j_{\theta_p} \in G_{\theta_p}\}$ is the set of edges, and $W = \{w_{p^i_{\theta_g}, p^j_{\theta_p}} : w_{p^i_{\theta_g}, p^j_{\theta_p}} = \gamma(p^i_{\theta_g}, p^j_{\theta_p})\}$ is the sets of edge-weights. The weight represents the correlation between two pixel-variables $(p^i_{\theta_g}, p^j_{\theta_p})$ from two different views (θ_g, θ_p) based on the absolute value of the correlation coefficient (γ) [2] as follow.

$$\gamma(p^i_{\theta_g}, p^j_{\theta_p}) = \left| \frac{\sum_{n=1}^{N_a} (p^i_{n,\theta_g} - \bar{p}^i_{\theta_g})(p^j_{n,\theta_p} - \bar{p}^j_{\theta_p})}{\sqrt{\sum_{n=1}^{N_a} (p^i_{n,\theta_g} - \bar{p}^i_{\theta_g})^2} \sqrt{\sum_{n=1}^{N_a} (p^j_{n,\theta_p} - \bar{p}^j_{\theta_p})^2}} \right| \tag{5.4}$$

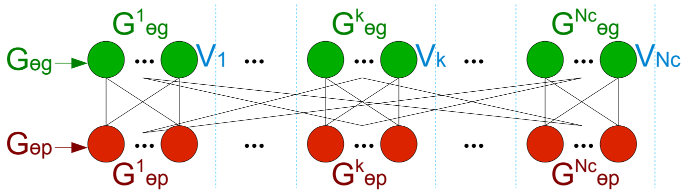


Figure 5.3: Graph partitioning. Green and red circles (i.e graph vertices) represent pixels in G_{θ_g} and G_{θ_p} respectively. Black lines are edges connecting each pixel in G_{θ_g} to each pixel in G_{θ_p}. Blue lines are graph cuts.

where p^i_{n,θ_g} and p^j_{n,θ_p} are values of the i^{th} and j^{th} pixels in *GEI* training samples G_{n,θ_g} and G_{n,θ_p} respectively, N_a is the total number of training samples, $\bar{p}^i_{\theta_g} = \frac{\sum_{n=1}^{N_a} p^i_{n,\theta_g}}{N_a}$, and $\bar{p}^j_{\theta_p} = \frac{\sum_{n=1}^{N_a} p^j_{n,\theta_p}}{N_a}$.

5.3.3 Bipartite graph multipartitioning on gait features from different views

Given the bipartite graph B on G_{θ_g} and G_{θ_p} from Section 5.3.2, *SVD* is applied to cluster the graph vertices (V) of B into N_c partitions as $V_1 = G^1_{\theta_g} \cup G^1_{\theta_p}, ..., V_{N_c} = G^{N_c}_{\theta_g} \cup G^{N_c}_{\theta_p}$ where $G^k_{\theta_g}$ and $G^k_{\theta_p}$ $(1 \le k \le N_c)$ are not necessary to have the same size. As shown in Figure 5.3, this process achieves *gait partitioning* simultaneously as $G_{\theta_g} = G^1_{\theta_g} \cup ... \cup G^{N_c}_{\theta_g}$ and $G_{\theta_p} = G^1_{\theta_p} \cup ... \cup G^{N_c}_{\theta_p}$.

The *gait partitioning* model can be obtained based on the *bipartite graph multipartitioning* which is explained in Section 2.6, through the following steps in Algorithm 1. First, the matrices $A \in \Re^{N_b \times N_d}$, $D_1 \in \Re^{N_b \times N_b}$, and $D_2 \in \Re^{N_d \times N_d}$ are constructed such that $A(i,j) = w_{p^i_{\theta_g}, p^j_{\theta_p}}$, D_1 and D_2 are diagonal matrices, $D_1(i,i) = \sum_{j=1}^{N_d} A(i,j)$, and $D_2(j,j) = \sum_{i=1}^{N_b} A(i,j)$. Then, the matrix A_n is formed based on these matrices A, D_1, and D_2.

As mentioned in Section 2.6, l left and right singular vectors ($\mathbf{U}$ and $\mathbf{V}$) of A_n contain N_c-modal information for N_c-partitioning (i.e. multipartitioning) where $l = \lceil log_2 N_c \rceil$. As defined herein, $\mathbf{U} = [u_2, ..., u_{l+1}] \in \Re^{N_b \times l}$ and $\mathbf{V} = [v_2, ..., v_{l+1}] \in \Re^{N_d \times l}$ where u_i and v_i are the left and right singular vectors respectively corresponding to the i^{th} largest singular value of A_n, and N_b and N_d are the dimensions of G_{θ_g} and G_{θ_p} respectively.

Algorithm 1 The *gait partitioning* model based on the *bipartite graph multipartitioning*

Input: A training dataset $\{G_{n,\theta_g}, G_{n,\theta_p}\}_{n=1}^{N_a}$ where each pair contains *GEI*s of the same subject from different views

Output: The *gait partitioning* model

1: Compute A, D_1 and D_2 based on the training dataset (see Equation (2.24))
2: Form $A_n = D_1^{-1/2} A D_2^{-1/2}$
3: Compute $l = \lceil log_2 N_c \rceil$ left and right singular vectors of A_n ($\mathbf{U} = (u_2, u_3, ..., u_{l+1})$ and $\mathbf{V} = (v_2, v_3, ..., v_{l+1})$)
4: Form $Z = \begin{bmatrix} D_1^{-1/2}\mathbf{U} \\ D_2^{-1/2}\mathbf{V} \end{bmatrix}$
5: Perform k-means ($k = N_c$) on the l-dimensional vectors in Z to obtain the desired N_c-way multipartitioning $\{G_{\theta_g}^k \cup G_{\theta_p}^k\}_{k=1}^{N_c}$
6: **return** Achieve the *gait partitioning* model as $G_{\theta_g} = G_{\theta_g}^1 \cup ... \cup G_{\theta_g}^{N_c}$ and $G_{\theta_p} = G_{\theta_p}^1 \cup ... \cup G_{\theta_p}^{N_c}$

At this stage, the eigenvectors (Z) which relax the *bipartite graph mutipartitioning* are obtained through the singular vectors in $\mathbf{U}$ and $\mathbf{V}$. Each l-dimensional vector in Z is a representative information of each vertex (which corresponds to each pixel in G_{θ_g} and G_{θ_p}) of the graph B. According to Equation (2.27) and the constructions of U and V, the first N_b vectors in Z correspond to pixels in G_{θ_g} and the rest of N_d vectors in Z correspond to pixels in G_{θ_p}.

Then, a simple clustering algorithm such as k-mean ($k = N_c$) can be applied on these l-dimensional vectors in Z to cluster them into N_c groups which sequentially provide the multipartitioning of the corresponding vertices (V) of the graph B ($\{G_{\theta_g}^k \cup G_{\theta_p}^k\}_{k=1}^{N_c}$) that agrees with our intuition in Equations (5.2) and (5.3). Accordingly, this results in the *gait partitioning* model as stated in Equation (5.1).

5.4 View-Normalization based on Correlation Optimization and Linear Approximation

It can be concluded from the proposed *gait partitioning* model that when *GEI*s are clustered into N_c segments, the corresponding segments $(G^k_{\theta_g}, G^k_{\theta_p})$ are the most correlated and likely to contain the same gait information but under different views (θ_g, θ_p). In this section, *CCA* (as briefed in Section 2.7) is applied to optimize the correlation of each pair of segments $(G^k_{\theta_g}, G^k_{\theta_p})$ where the size of $G^k_{\theta_g}$ (N_g) and the size of $G^k_{\theta_p}$ (N_p) are not necessarily the same as mentioned above.

5.4.1 Correlation optimization

CCA has the ability to model the relationships between two sets of variables (i.e. $G^k_{\theta_g}$ and $G^k_{\theta_p}$ in this study). It also performs the data reduction in this multivariate analysis. In this chapter, *CCA* is applied to compute a pair of projection vectors w^k_g and w^k_p such that the correlation ρ between the two projections $(w^k_g)^T G^k_{\theta_g}$ and $(w^k_p)^T G^k_{\theta_p}$ is maximized. $w^k_g \in \Re^{N_g \times 1}$ and $w^k_p \in \Re^{N_p \times 1}$ are eigenvectors of the following eigenvalue equations.

$$\begin{aligned} C^{-1}_{gg}\ C_{gp}\ C^{-1}_{pp}\ C_{pg}\ \omega^k_g &= \rho^2 \omega^k_g \\ C^{-1}_{pp}\ C_{pg}\ C^{-1}_{gg}\ C_{gp}\ \omega^k_p &= \rho^2 \omega^k_p \end{aligned} \tag{5.5}$$

where Cgg and Cpp are the within-set covariance matrices of $G^k_{\theta_g}$ and $G^k_{\theta_p}$ respectively, and Cgp is the between-set covariance matrix of $G^k_{\theta_g}$ and $G^k_{\theta_p}$.

In this study, for the subspace optimization, we use *CCA* projection matrices ($\mathbf{W}^k_g$ and $\mathbf{W}^k_p$) which contain the first N_e eigenvectors where their corresponding eigenvalues are approximately 1. As defined herein, $\mathbf{W}^k_g = \{w^k_{g,i}\}^{N_e}_{i=1} \in \Re^{N_g \times N_e}$ and $\mathbf{W}^k_p = \{w^k_{p,i}\}^{N_e}_{i=1} \in \Re^{N_p \times N_e}$ where $w^k_{g,i}$ and $w^k_{p,i}$ is the i^{th} pair of eigenvectors corresponding to the i^{th} largest eigenvalue (ρ_i) of Equation (5.5), and $\rho_i \approx 1$.

The *CCA* projection matrices are used to maximize the correlation between $G^k_{\theta_g}$ and $G^k_{\theta_p}$. They are also used to reduce the gait feature dimensions (i.e. $N_e < N_g$ and $N_e < N_p$).

Therefore, $G^k_{\theta_g} \in \Re^{N_g}$ and $G^k_{\theta_p} \in \Re^{N_p}$ can be projected into *CCA* subspaces as $\tilde{G}^k_{\theta_g} = (\mathbf{W}^k_g)^T G^k_{\theta_g} \in \Re^{N_e}$ and $\tilde{G}^k_{\theta_p} = (\mathbf{W}^k_p)^T G^k_{\theta_p} \in \Re^{N_e}$ respectively.

5.4.2 Linear approximation

Let $a_i^k \in \Re$ and $b_i^k \in \Re$ be the projections of $G_{\theta_g}^k$ and $G_{\theta_p}^k$ through the projection vectors $\omega_{g,i}^k$ and $\omega_{p,i}^k$ respectively. Thus,

$$a_i^k = (\omega_{g,i}^k)^T G_{\theta_g}^k, \;\; b_i^k = (\omega_{p,i}^k)^T G_{\theta_p}^k \tag{5.6}$$

Suppose that the corresponding correlation value ρ_i is approximately 1, then a_i^k and b_i^k are approximately linearly correlated. Thus, there are coefficient β_i^k and constant ϵ_i^k to define the linear equation as follow.

$$a_i^k = \beta_i^k b_i^k + \epsilon_i^k \tag{5.7}$$

Given the training dataset $\{G_{n,\theta_g}, G_{n,\theta_p}\}_{n=1}^{N_a}$ (where each pair contains *GEI* samples of the same subject from different views, G_{n,θ_g} and G_{n,θ_p} are the n^{th} samples of G_{θ_g} and G_{θ_p} respectively, and N_a is the total number of training samples), β_i^k and ϵ_i^k can be calculated according to the regression analysis [133] as follow.

$$\begin{aligned}
\beta_i^k &= \frac{\sum_{n=1}^{N_a} (\omega_{p,i}^k)^T G_{n,\theta_p}^k (\omega_{g,i}^k)^T G_{n,\theta_g}^k}{\sum_{n=1}^{N_a} ((\omega_{p,i}^k)^T G_{n,\theta_p}^k)^2} \\
\epsilon_i^k &= \frac{1}{N_a} \sum_{n=1}^{N_a} (\omega_{g,i}^k)^T G_{n,\theta_g}^k - \frac{\beta_i^k}{N_a} \sum_{n=1}^{N_a} (\omega_{p,i}^k)^T G_{n,\theta_p}^k
\end{aligned} \tag{5.8}$$

Therefore, $\vec{\beta^k} = \{\beta_i^k\}_{i=1}^{N_e}$ and $\vec{\epsilon^k} = \{\epsilon_i^k\}_{i=1}^{N_e}$ can be used to linearly transform *GEI* from θ_p to θ_g under the trained *CCA* subspaces ($\tilde{G}_{\theta_p \to \theta_g}^k = \vec{\beta^k} \tilde{G}_{\theta_p}^k + \vec{\epsilon^k}$). In this way, two sets of gaits previously under different views are now transformed onto a same view where the gait similarity measurement can be carried out.

5.5 Gait Recognition under Various Views

Two scenarios of gait recognition under various views are addressed as follows.

Algorithm 2 Cross-view gait similarity measurement

Input: Two *GEI*s from different views (G_{1,θ_g} and G_{2,θ_p})

Output: The gait similarity

1: *GEI*s are co-clustered into N_c segments based on the trained *GEI* partitioning model:
$G_{1,\theta_g} = \{G^k_{1,\theta_g}\}^{N_c}_{k=1}$ and
$G_{2,\theta_p} = \{G^k_{2,\theta_p}\}^{N_c}_{k=1}$

2: *GEI*s are transformed into the trained *CCA* subspaces:
$\tilde{G}_{1,\theta_g} = \{\tilde{G}^k_{1,\theta_g}\}^{N_c}_{k=1} = \{(\mathbf{W}^k_g)^T G^k_{1,\theta_g}\}^{N_c}_{k=1}$ and
$\tilde{G}_{2,\theta_p} = \{\tilde{G}^k_{2,\theta_p}\}^{N_c}_{k=1} = \{(\mathbf{W}^k_p)^T G^k_{2,\theta_p}\}^{N_c}_{k=1}$

3: *GEI* from θ_p is transformed to θ_g under the trained *CCA* subspaces:
$\tilde{G}_{2,\theta_p\to\theta_g} = \{\tilde{G}^k_{2,\theta_p\to\theta_g}\}^{N_c}_{k=1} = \{\vec{\beta^k}\tilde{G}^k_{2,\theta_p} + \vec{\epsilon^k}\}^{N_c}_{k=1}$

4: **return** Gait similarity is a sum of cosine similarities of corresponding segments:
$sim = \sum_{k=1}^{N_c} \frac{\tilde{G}^k_{2,\theta_p\to\theta_g} \cdot \tilde{G}^k_{1,\theta_g}}{||\tilde{G}^k_{2,\theta_p\to\theta_g}|| \;\; ||\tilde{G}^k_{1,\theta_g}||}$

5.5.1 Cross-view gait recognition

Based on the training processes in Sections 5.3 and 5.4, the view-normalization model for θ_g and θ_p can be defined as follow.

$$M_{\theta_g,\theta_p} = \{\ G^k_{\theta_g},\ G^k_{\theta_p},\ \mathbf{W}^k_g,\ \mathbf{W}^k_p,\ \vec{\beta^k},\ \vec{\epsilon^k}\ \}^{N_c}_{k=1} \tag{5.9}$$

In the cross-view recognition phase, given any two testing samples G_{1,θ_g} and G_{2,θ_p}, gait similarity is measured by following the steps in Algorithm 2 based on the trained M_{θ_g,θ_p}. The higher sim, the more possibility that the two gaits $(G_{1,\theta_g}, G_{2,\theta_p})$ belong to the same subject.

5.5.2 Multi-view gait recognition

As mentioned in the introduction, *multi-view gait recognition* is defined as the scenario where there are at least two views available for gallery and/or probe gaits. For example, there are A views available for gallery gait $(G_{\theta_{g1}}, G_{\theta_{g2}}, ..., G_{\theta_{gA}})$

and B views available for probe gait $(G_{\theta_{p1}}, G_{\theta_{p2}}, ..., G_{\theta_{pB}})$, of each person. Thus, gallery and probe gait feature vectors are defined as follow.

$$
\begin{aligned}
G_{\theta_A} &= G_{\theta_{g1}} : G_{\theta_{g2}} : ... : G_{\theta_{gA}} \\
G_{\theta_B} &= G_{\theta_{p1}} : G_{\theta_{p2}} : ... : G_{\theta_{pB}}
\end{aligned}
\tag{5.10}
$$

where ':' is the concatenation. Then, the same method for *cross-view gait recognition* (Sections 5.3, 5.4, and 5.5.1) is applied for *multi-view gait recognition* given the extended feature vectors described in Equation (5.10). That is, G_{θ_A} and G_{θ_B} will be in place of G_{θ_g} and G_{θ_p} respectively in Algorithm 2.

5.6 Experiments

The CASIA gait database B is used in our experiments (Sections 5.6.1, 5.6.2, and 5.6.3). This database directly supports the study of gait recognition under view change, which contains 11 views (0°, 18°, ..., 180°). The database is randomly divided into two groups. The first group of 24 subjects are used for the training processes: *gait partitioning* model, correlation optimization, and *linear approximation* (see Sections 5.3 and 5.4). The rest of 100 subjects are used for evaluating the performance of gait recognition under view change. Therefore, the training processes are independent from the subject recognition process. It demonstrates generality of the trained models.

Moreover, our experiment in Section 5.6.4 is carried out based on the USF gait database which is considered as a real scene. This database is challenging because of several difficulties of the outdoor environment such as wind, shadow, and illumination. It is recorded from two views/cameras (L and R). The cameras' lines of sight are verged at approximately 30 degrees. In our experiment, 22 subjects are randomly selected and used for the training processes. The rest of 100 subjects are used for evaluating *cross-view gait recognition*.

In our experiments, the proposed method is implemented using the library functions of OpenCV 2.0 in Microsoft Visual C++ 8.0 environment on the computer with Quad Processor 2.66 GHz and 4 GB Ram. Regarding the computational complexity, the training processes may take time but the view-transformation

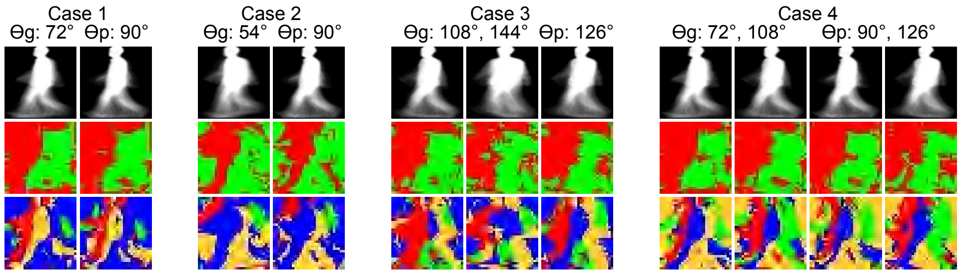


Figure 5.4: Sample results of *GEI* partitioning model. *GEI*s are clustered into 2 and 4 segments as shown in rows 2 and 3 respectively. Row 1 shows sample *GEI*s from different views. Different colors are used to distinguish different segments.

process itself in the recognition phase is very fast. In our experiment, it took approximately 10 minutes to train one view-normalization model but it took only less than 0.001 second to achieve one view-transformation. The training processes can be completed offline beforehand.

5.6.1 GEI partitioning model

This section is to illustrate sample results of our *GEI* partitioning model proposed in Section 5.3. From Figure 5.4, cases 1 and 2 are *cross-view gait recognitions* when the view-differences are 18° and 36° respectively, case 3 is *multi-view to one-view gait recognition*, and case 4 is *multi-view to multi-view gait recognition*.

From Figure 5.4, several important points can be observed. First, pixels representing the similar motion regions are clustered into the same segment. For example, from the third row in case 1, yellow segments correspond to body parts that move less during a *walking cycle* (i.e. head, torso, middle side of legs), red and green segments correspond to body parts that move constantly during a *walking cycle* (i.e. arms, outer side of legs), and blue segments correspond to background and stride (i.e. movements between legs).

Second, the corresponding segments from different views are likely to contain correlated gait information from the same body parts. For example, from the third row in case 4, red and green segments represent the front-most and back-most parts respectively of human body, which contain correlated gait information across views. Although these segments from different views may include gait information of different body parts, these body parts are still related to some extent (e.g. different areas of the same 3D leg) depending on the degree of view change. Moreover, blue segments represent the middle parts of human body, which usually contain shared gait information across views. Yellow segments represent strides and backgrounds captured from different views.

Third, the corresponding segments (i.e. segments with the same color in Figure 5.4) from different views are allocated in different x-y positions of *GEI*, although they are likely to contain the same body part as mentioned above. This is because they belong to the different geometric representations of different views.

Based on the above observations, applying *CCA* on the corresponding segments instead of the global gait features (i.e. the proposed method) can avoid spurious relationships between unrelated motions.

5.6.2 Cross-view gait recognition

The performance on *cross-view gait recognition* proposed in Section 5.5.1 is evaluated and compared with other six methods (see Table 5.2) including: 1) the baseline method which simply matches *GEI* across views without any view-normalization [3]; 2) view rectification using self-calibrating [44]; 3) FT-*dVTM* [52]; 4) *GEI-dVTM* [50]; 5) *SVR-rVTM* [49]; 6) *GEI-CCA* [48]. These methods [3][44][48][49][50][52] are the key methods of gait recognition under view change. In Table 5.2, some missing results have not been reported by [44]. In our experiments, Nearest Neighbor (NN) is used as a classifier for gait recognition.

In this section, the experimental results corresponding to four probe views ($0°$, $54°$, $90°$, $126°$) are shown in Table 5.2. These views are selected to represent the common situations: frontal view ($0°$), oblique view ($54°$, $126°$), and literal view ($90°$). In our experiments, it was observed that the performances for the rest of probe views are similar as shown in Table 5.2.

In this experiment, for a fair comparison, the related method in [48] is reimplemented and adjusted to be consistent with the proposed method in the following aspects: 1) gait feature; 2) gait similarity measurement; 3) training and testing datasets. In this way, advantages of the proposed *gait partitioning* model can be clearly demonstrated. From Table 5.2, it is clearly seen that our method outperforms the method in [48] especially for the case of large view change. For example, when compared with [48], our method improves the performance averagely by 7% and 12% for ± 18 and ± 36 degrees of view changes respectively.

Among the *VTM*-based methods [48][49][50], the method in [49] provides the highest accuracy. However, it performs worse than our method in all cases (especially for the case of large view change). This is because our method relies on the more robust segment to segment mapping on the optimized subspaces. The performance improvement of the proposed method against [49] is further demonstrated in *multi-view gait recognition* (see Section 5.6.3).

Moreover, our method significantly outperforms the baseline method [3]. It also performs better than [44] for most cases except some cases of relatively large view-difference. In our method, this limitation of large view-difference can be overcome based on *multi-view gait recognition* (see Sections 5.5.2 and 5.6.3) which has not been discussed by [44].

For example, in the case of $\theta_g = 54°, \theta_p = 90°$ (±36 degrees of view change) from Table 5.2, [44] achieves 74% which is better than 66% achieved by the proposed method. However, *multi-view gait recognition* in the case of $\theta_g = 54°, \theta_p = 18°, 90°$ (±36 degrees of view change) from Table 5.3 shows that the proposed method achieves the improved recognition performance of 83%. Also, the method in [44] cannot be efficiently applied for frontal views (0°,18°,144°,162°,180°) because limbs' poses (i.e. gait information used in [44]) become unreliable and/or untraceable under these views.

Furthermore, it is clearly seen that *cross-view gait recognition* between two closer views achieves better performance because the gaits share more information when their views are closer.

5.6.3 Multi-view gait recognition

In Section 5.5.2, it shows the capability of the proposed method in a multi-camera environment but without camera cooperation. Two scenarios are demonstrated here: 1) *multi-view to one-view gait recognition* where gallery gaits from multiple views are used to recognize probe gait from one view; 2) *multi-view to multi-view* *gait recognition* where gallery gaits from multiple views are used to recognize probe gaits from multiple views. These scenarios are quite common in the real world, where gallery and/or probe gaits from multiple views are available.

5.6.3.1 Multi-view to one-view gait recognition

Table 5.3 shows that the performance of *two-view to one-view gait recognition* is significantly better than *cross-view gait recognition*. The reason is because gaits from multiple views provide more information for view-normalization process.

Moreover, based on our literature review, there are two existing methods [49][50] applicable for this scenario of *multi-view to one-view gait recognition*. They are used in the comparison. It can be seen that the proposed method outperforms the *VTM*-based methods [49][50].

5.6.3.2 Multi-view to multi-view gait recognition

Table 5.4 shows the experimental results and comparisons with the method in [48] which is the only existing method that may address this same scenario. Our method is the first time to introduce *multi-view to multi-view gait recognition* without camera cooperation. To make a comparison, the method in [48] is reimplemented as mentioned in Section 5.6.2. For more comprehensive comparisons, the relevant experimental results of *two-view to one-view* and *cross-view gait recognitions* are also shown as the references.

From Table 5.4, *two-view to two-view gait recognition* achieves the highest accuracy, followed by *two-view to one-view gait recognition* and *cross-view gait recognition* respectively. It also can be seen that the proposed method clearly outperforms the method in [48].

5.6.4 Cross-view gait recognition in a cluttered outdoor environment

The proposed method is further evaluated using the practical dataset (outdoor) i.e. the USF gait database. Among the 12 challenging experiments pre-defined for the USF gait database, the experiment A is adopted for our evaluation because it observes view change for gait recognition which matches the focus of this study. Probe and gallery gaits are recorded from different cameras L and R respectively.

From Table 5.5, the proposed method (which achieves 89%) outperforms other methods in the literature (which achieve below 85%).

5.7 Conclusion

In this chapter, a new method has been proposed to address the gait recognition under view change. Gallery and probe gaits from different views are co-clustered into segments such that the corresponding segments from different views are the most correlated. To bridge the gap across gaits from different views, *CCA* is applied on these corresponding segments to maximize their linear correlation. As a result, gallery and probe gaits in the *CCA* transformed subspaces are approximately linearly correlated. Then, gait similarity measurement is carried out based on *linear approximation* and cosine similarity. The comprehensive experimental results have shown that our method outperforms other methods in the literature.

To improve the gait recognition performance, the proposed framework for *cross-view gait recognition* has been further applied for *multi-view gait recognition* without camera cooperation. In our experiments, it has been also shown that *multi-view gait recognition* performs better than *cross-view gait recognition* in all cases. Besides, our method outperforms other methods for the same scenario of *multi-view gait recognition*.

In the future work, a non-linear modelling will be attempted. The correlation between gaits from different views will be modelled using a non-linear relationship, and a non-linear correlation optimization will be then applied. Moreover, we will also consider to use other types of gait features that may provide more useful and reliable information for recognizing gaits across views, based on the proposed method.

Table 5.2: Comparisons on *cross-view gait recognition* (%) using different methods.

Probe view (θ_p)	0°									
Gallery view (θ_g)	18°	36°	54°	72°	90°	108°	126°	144°	162°	180°
Baseline [3]	24	4	2	1	1	2	1	4	12	41
View-rectification [44]	–	–	–	–	–	–	–	–	–	–
FT-*dVTM* [52]	26	9	4	5	3	5	6	9	19	42
GEI-dVTM [50]	69	33	12	11	9	11	15	19	45	67
SVR-rVTM [49]	84	45	23	23	25	21	23	29	67	94
GEI-CCA [48]	77	39	19	16	19	19	20	29	62	91
The proposed method	**85**	**47**	**26**	**25**	**28**	**25**	**27**	**37**	**68**	**95**

Probe view (θ_p)	54°									
Gallery view (θ_g)	0°	18°	36°	72°	90°	108°	126°	144°	162°	180°
Baseline [3]	4	9	30	22	18	17	38	19	2	3
View-rectification [44]	–	–	57	65	62	**63**	**63**	–	–	–
FT-*dVTM* [52]	8	31	72	43	28	19	24	18	14	12
GEI-dVTM [50]	13	46	87	81	49	31	27	19	18	16
SVR-rVTM [49]	22	64	95	93	59	51	42	27	20	21
GEI-CCA [48]	22	56	94	88	51	47	43	27	17	15
The proposed method	**24**	**65**	**97**	**95**	**63**	53	48	**34**	**23**	**22**

Probe view (θ_p)	90°									
Gallery view (θ_g)	0°	18°	36°	54°	72°	108°	126°	144°	162°	180°
Baseline [3]	3	4	7	17	82	88	22	2	1	1
View-rectification [44]	–	–	**53**	**74**	73	69	67	–	–	–
FT-*dVTM* [52]	3	6	8	27	36	58	28	21	10	3
GEI-dVTM [50]	7	11	22	52	75	79	45	26	12	6
SVR-rVTM [49]	16	22	35	63	95	**95**	65	38	20	**13**
GEI-CCA [48]	12	18	30	55	89	91	51	32	16	8
The proposed method	**18**	**24**	41	66	**96**	**95**	**68**	**41**	**21**	**13**

Probe view (θ_p)	126°									
Gallery view (θ_g)	0°	18°	36°	54°	72°	90°	108°	144°	162°	180°
Baseline [3]	2	5	13	29	21	15	37	43	2	4
View-rectification [44]	–	–	**45**	**57**	**60**	70	68	–	–	–
FT-*dVTM* [52]	4	6	7	17	16	22	55	76	26	8
GEI-dVTM [50]	16	17	21	31	42	53	80	95	49	14
SVR-rVTM [49]	22	26	26	42	57	**78**	**98**	**98**	74	19
GEI-CCA [48]	21	18	26	34	54	75	91	97	63	18
The proposed method	**25**	**29**	35	49	**60**	**78**	**98**	**98**	**75**	**22**

Table 5.3: Comparisons on *multi-view to one-view gait recognition* (%) using different methods.

Scenario	two-view to one-view	cross-view	cross-view
Gallery view (θ_g)	$54°$		
Probe view (θ_p)	$36°, 72°$	$36°$	$72°$
GEI-dVTM [50]	95	81	85
SVR-rVTM [49]	**99**	**92**	90
The proposed method	**99**	**92**	**92**
Gallery view (θ_g)	$54°$		
Probe view (θ_p)	$18°, 90°$	$18°$	$90°$
GEI-dVTM [50]	74	34	52
SVR-rVTM [49]	80	46	63
The proposed method	**83**	**53**	**66**
Gallery view (θ_g)	$54°$		
Probe view (θ_p)	$0°, 108°$	$0°$	$108°$
GEI-dVTM [50]	40	12	38
SVR-rVTM [49]	54	23	42
The proposed method	**57**	**26**	**44**
Gallery view (θ_g)	$126°$		
Probe view (θ_p)	$108°, 144°$	$108°$	$144°$
GEI-dVTM [50]	97	84	92
SVR-rVTM [49]	98	**96**	**97**
The proposed method	**99**	**96**	**97**
Gallery view (θ_g)	$126°$		
Probe view (θ_p)	$90°, 162°$	$90°$	$162°$
GEI-dVTM [50]	72	45	50
SVR-rVTM [49]	88	65	53
The proposed method	**90**	**68**	**59**
Gallery view (θ_g)	$126°$		
Probe view (θ_p)	$72°, 180°$	$72°$	$180°$
GEI-dVTM [50]	35	31	16
SVR-rVTM [49]	54	45	**34**
The proposed method	**60**	**47**	**34**

Table 5.4: Comparisons on *multi-view to multi-view gait recognition* (%) using different methods.

Scenario	two-view to two-view	two-view to one-view	cross-view
Gallery view (θ_g)	$54°, 90°$	$54°, 90°$	$90°$
Probe view (θ_p)	$72°, 108°$	$72°$	$72°$
GEI-CCA [48]	96	96	89
The proposed method	**99**	**98**	**96**
Gallery view (θ_g)	$72°, 108°$	$72°, 108°$	$108°$
Probe view (θ_p)	$90°, 126°$	$90°$	$90°$
GEI-CCA [48]	97	96	91
The proposed method	**100**	**98**	**95**
Gallery view (θ_g)	$18°, 90°$	$18°, 90°$	$90°$
Probe view (θ_p)	$54°, 126°$	$54°$	$54°$
GEI-CCA [48]	83	73	51
The proposed method	**89**	**80**	**63**
Gallery view (θ_g)	$36°, 108°$	$36°, 108°$	$108°$
Probe view (θ_p)	$72°, 144°$	$72°$	$72°$
GEI-CCA [48]	79	74	58
The proposed method	**85**	**79**	**67**

Table 5.5: Performance of cross-view gait recognition (%) in a cluttered outdoor environment using the experiment A of the USF gait database.

Method	Recognition rate (%)
GEI [27]	83
MSCT+SST [130]	80
pHMM [28]	85
PEI+*LDA* [29]	85
USF [2]	73
HMM [132]	80
CFET [129]	83
The proposed method	**89**

Chapter 6

Gait Recognition Across Various Walking Speeds Using Higher Order Shape Configuration

6.1 Introduction

In a real-world environment, there are various factors significantly affecting human gait including dressing in different clothes, walking while carrying different objects, walking on different surfaces, walking with different shoes, walking under variable speeds, and walking being observed from arbitrary views [2]. Among these factors, speed change has been regarded as one of most commonly seen challenging factors. The change of walking speed can significantly change the gait shape description. For example, the reduction of stride length can be caused by slowing down the walking speed [14]. In fact, speed change also affects other body parts including arm, hip, knee, and ankle. Moreover, different people may react differently to the change of walking speed because of age, gender, individual body structure, etc., which makes the situation more complicated. This chapter will focus on effects of speed change on gait recognition. A new solution is proposed to deal with the challenges raised.

In recent years, several methods (see Section 1.2.3) have been proposed for cross-speed gait recognition from different perspectives. Based on their published results, it is observed that the problem can be resolved to some extent, particularly, when the speed changes are small. However, it is still challenging when the speed changes are significant. The proposed method will tackle this problem as one of the core research objectives in this chapter. To demonstrate the performance improvements by the proposed method, the relevant comparisons have been conducted in this study.

In this chapter, *Procrustes Shape Analysis (PSA)* is adopted for gait recognition because it has been proved as a special shape description which can tolerate the change of orientation of an object [106][134][135]. It can help to conduct shape registrations, in particular, to build up gait features. Somehow, *PSA* can deal with the changes of gait pose and body size throughout the *walking cycle* although its original scheme is still constrained to cross-speed gait recognition. The main challenges raised in the current *PSA* framework are: 1) the existing shape configuration method is based on the static shape description known as *Centroid Shape Configuration (CSC)*. It may not efficiently describe dynamic gait information particularly with the variation of walking speed; 2) the existing similarity measurement is carried out between two shapes as a whole without differentiating significances between different segments of the shape. It will cause problems if contributions of the different shape segments are not the same.

In order to adopt the *PSA* and to adapt it for our purpose, we carry out a complete analysis on details of effects caused by various walking speed changes on gait recognition. Moreover, we also demonstrate the various effects on different body parts. Such analysis provides helpful references for future work in this area. According to the analysis, this chapter proposes a new *Higher-order Shape Configuration (HSC)* which satisfactorily extends traditional *CSC* to handle the gait shape change caused by the walking speed change. In the meantime, we introduce *Differential Composition Model (DCM)* which reflects different effects caused by walking speed change on the different body parts. *DCM* further improves the discriminability of the gait feature based on *HSC*. Based on our experiments, the proposed method has achieved a much better performance when compared with the benchmark methods, particularly, when the change of walking speed is significant.

In summary, the proposed *PSA*-based method for speed-invariant gait recognition has the following significant points which are different from the traditional *PSA* framework [90][92][106][134][135][136][137][138][139].

- In the proposed method for pre-processing, prior knowledge of human shape structure is embedded in the re-sampling process to more precisely address the point correspondences of shape configurations. Three key positions of the human body (i.e. head, left foot, and right foot) are automatically detected and adopted as the reference points for the re-sampling. This assumption is reasonable for gait shape analysis since they are visible normally in a standing posture under any speed and view.
- To describe gait shape, *HSC* is proposed to replace the traditional *CSC* in the *PSA* scheme which is not efficient in handling the speed change problem. *HSC* describes gait shape using higher-order information of the shape boundary such as tangent and curvature. Such information is consistent in describing gait shape regardless of any global appearance change caused by speed change. The Forward Divided Difference approximation is applied for calculating higher-order derivatives [140].
- In order to handle the large speed change, *DCM* is proposed to decompose the gait shape boundary into segments. Such segmentation reflects various efforts caused by speed changes on different body parts. Each segment is assigned a weight to differentiate itself. Fisher discriminant is used to calculate the weight values. Then, the final similarity measurement between any two gaits is a weighted sum of the distances of each corresponding pair of boundary segments.

The rest of this chapter is organized as follows. Pre-processing is explained in Section 6.2. The framework of *PSA* for gait recognition is introduced in Section 6.3. Effects of walking speed change on the *PSA*-based gait recognition are comprehensively analyzed in Section 6.4, followed by the proposed method of cross-speed gait recognition based on the improved scheme of *PSA* in Section 6.5. Experimental results are shown in Section 6.6, and conclusions are drawn in Section 6.7.

6.2 Pre-processing and Walking Speed Estimation

As similar to other gait recognition methods, gait should be analyzed within complete *walking cycle*(s) because it is a periodic action. The method for *gait period* estimation is proposed in Section 3.2. A gait silhouette is extracted from each frame as explained in Section 3.2. Then, a shape boundary is obtained from each gait silhouette by using the Border Following algorithm [141].

In practice, gait shapes of an individual can vary in size and pose. For example, gait shapes of a leg-crossing pose and a leg-apart pose are very different, and gait shapes usually appear larger when they are closer to a camera. Thus, a process of shape re-sampling is required for shape normalization and point correspondence between different gait shape boundaries. So that, all re-sampled shape boundaries will contain the same number of boundary points (N_p) and each boundary point approximately corresponds to the same body part. This is important pre-processing in our gait recognition because: 1) *PSA* is rigid evaluations and requires a one-to-one point correspondence between the shapes; 2) gait is a dynamic shape model such that it varies in pose and size throughout a *walking cycle* and camera's viewpoint; and 3) particularly, gait shape changes when walking speed changes (see *Analysis-A*).

Selecting the proper number of re-sampling points (N_p) is essential. The larger number, the more details of shape are presented. However, it might increase computational complexity and introduce more possible noises around the shape boundary. On the other hand, the smaller number, the less details of shape are presented. Insufficient details can lead to a possible poor performance for shape analysis. In practice, N_p is usually determined empirically in different cases based on a training dataset.

In our study, prior knowledge of the human shape structure is integrated into the re-sampling process in order to improve the accuracy of point correspondences between different gait shapes. Based on the prior knowledge, the prominent positions of left foot, right foot, and head are used as the key reference points to provisionally divide the gait shape boundary into three clockwise curves: head−right

Table 6.1: Estimation of *gait period* for the CASIA gait database C.

Speed	T_{frame}	T_{second}
fs (slow speed)	29	1.16
fn (normal speed)	25	1.00
fq (fast speed)	22	0.88

foot, right foot−left foot, and left foot−head. These three points can be automatically estimated based on the analysis of projection histogram on major axis [142].

Then, the equal arc-length sampling [143] is applied independently on each boundary segment instead of a whole shape boundary. The equal arc-length sampling selects key points with the interval of equal arc length along the shape boundary. The space between two consecutive key points is given by L/K, where L is the perimeter of the boundary and K is the total number of re-sampled points. The optimized total numbers of re-sampled points on each segment are empirically determined by maximizing the recognition performance on the training dataset. In our experiment, each boundary segments are tested with five different values of K (i.e., 5, 10, 15, 20, and 25). The optimal combination is 15, 10, and 15 points for the boundary segments of head−right foot, right foot−left foot, and left foot−head, respectively. For further analyses, the re-sampled gait shape boundary will be reconsidered as a whole. In the rest of this chapter, boundary points mean the points identified by the re-sampling process above along the gait shape boundary.

After the re-sampling process, any gait shape boundary is represented as $\{(x_i, y_i)\}_{i=1}^{N_p}$ where (x_i, y_i) is x- and y-coordinates of the i^{th} re-sampled boundary point.

Moreover, since the objective of this study is to address the problem of walking speed change for gait recognition, it is necessary to know a walking speed of each gait sequence. The rest of this section proposes a reasonable way to automatically estimate the walking speed.

Table 6.2: Estimation of *gait period* for the OU-ISIR gait database.

Speed (km/h)	T_{frame}	T_{second}
2	105	1.75
3	81	1.35
4	68	1.13
5	60	1.00
6	54	0.90
7	49	0.82

Usually, walking speed is measured in a unit of km/h or m/s. Alternatively, it can be approximately measured as the period (seconds) of one gait cycle. A shorter period represents a higher walking speed. Tables 6.1 and 6.2 show the estimated gait periods under the various walking speeds based on the CASIA gait database C and the OU-ISIR gait database respectively. The *gait period* in a unit of frame number (T_{frame}) is estimated based on the method explained in Section 3.2.

When a video frame rate is given, the *gait period* in a unit of time e.g. second (T_{second}) is calculated as follow.

$$T_{second} = \frac{T_{frame}}{frame\ rate\ (f/s)} \tag{6.1}$$

The CASIA gait database C was recorded at 25 f/s, while the OU-ISIR gait database was recorded at 60 f/s. The CASIA gait database C does not provide any record of absolute walking speeds (km/h). However, they can be reasonably estimated using the OU-ISIR gait database as a reference based on T_{second} (see the 3^{rd} columns in Tables 6.1 and 6.2). Therefore, fs, fn and fq in the CASIA gait database C are approximately 4, 5 and 6 km/h respectively.

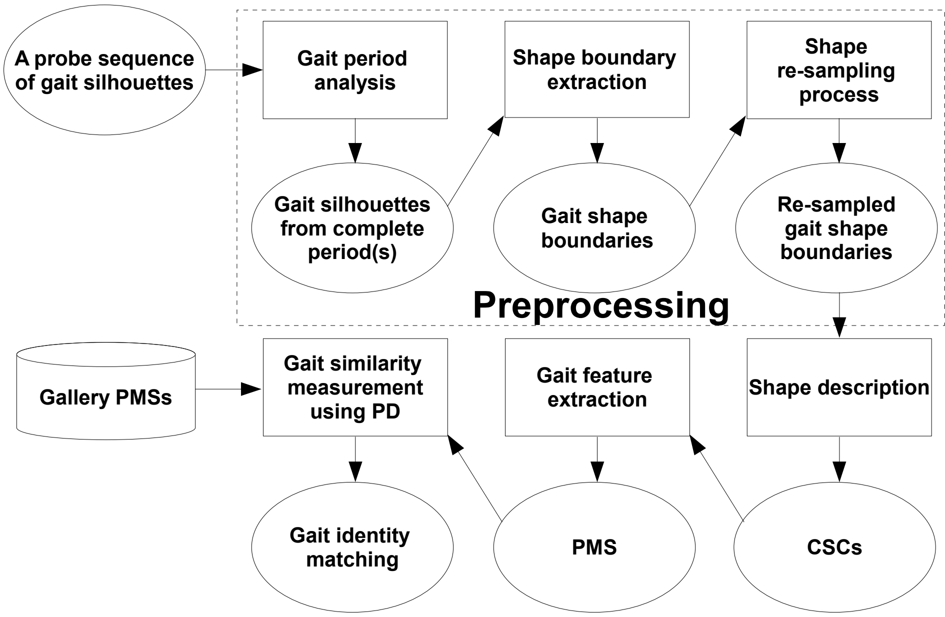


Figure 6.1: The framework of *PSA*-based gait recognition.

6.3 PSA-based Gait Recognition

First of all, in this section, the standard scheme of *PSA* adopted for gait recognition is introduced in order to demonstrate the challenges of walking speed change. The framework of *PSA*-based gait recognition is shown in Figure 6.1.

After gait shapes have been properly re-sampled as explained in Section 6.2, *CSC* is used to describe the normalized gait shapes. It describes the shape by recording the displacement of each boundary point from the shape centroid. Then, Procrustes Mean Shape (PMS) of a set of *CSC*s in complete *walking cycle*(s) is computed as a gait feature. The construction of PMS is based on a procedure of superimposition along a set of shapes by considering affine transformation (see Equation (2.31)) among them. Equation (2.31) is then extended for PMS construction by superimposing of multiple shapes. Finally, Procrustes Distance (PD) (see Equation 2.33) is subsequently used to measure dissimilarity between two PMSs of any two gait sequences.

In order to have a detailed understanding on the *PSA*-based gait recognition, brief discussions of the key components are given below.

6.3.1 CSC-based shape descriptor and its constraint to walking speed change

In the conventional framework [90][106][144], the re-sampled shape boundary is described using *CSC*. *CSC* is related to $r - s$ Curve that stores the distance r from each boundary point s to origin of the shape which is the shape centroid. By unwrapping the shape boundary into a set of boundary points, *CSC* can be described as a vector of complex numbers as follow.

$$Z = \{z_i | i = 1, 2, ..., N_p\}^T \tag{6.2}$$

where $z_i = (x_i - x_c) + j * (y_i - y_c)$, (x_i, y_i) is the i^{th} boundary point, (x_c, y_c) is the shape centroid, $x_c = \frac{\sum_{i=1}^{N_p} x_i}{N_p}$, $y_c = \frac{\sum_{i=1}^{N_p} y_i}{N_p}$, and N_p is the total number of boundary points. *CSC* is a global shape descriptor using the shape centroid as a global reference. The shape centroid is utilized as the origin of the 2-D shape space to register all shapes to a common center, which can handle translation invariance.

However, *CSC* has some disadvantages due to its global representation. In practice, gait shape of individual can be easily altered by many factors, particularly by the change of walking speed and the inconsistency of walking pattern of the individual. Therefore, position of the shape centroid is not stable. Furthermore, according to our experiments, *CSC* which describes the shape based on its global shape appearance, is very sensitive to walking speed change. In this chapter, *HSC* is proposed to replace *CSC* for describing gait shape.

6.3.2 PMS-based gait feature

Given a set of N_g gait shape configurations e.g. *CSC*s ($\{Z_i\}_{i=1}^{N_g}$) from complete *walking cycle*(s), PMS is estimated by extending the objective function in Equation (2.31), which minimizes a sum of PDs between the mean shape (Z_G i.e. PMS) and each gait shape configuration (Z_i) as follow.

$$\min_{\alpha_i,\ \beta_i} \sum_{i=1}^{N_g} ||Z_G - \alpha_i 1_k - \beta_i Z_i||^2, \qquad \beta_i = |\beta_i| e^{j \angle \beta_i} \tag{6.3}$$

where $\alpha_i 1_k$ gives the translation of Z_i, and $|\beta_i|$ and $\angle\beta_i$ give the scale and the rotation of Z_i, respectively.

Given Z_i are invariant to translation (e.g. gait shapes are registered using their shape centroids), the solution of Equation (6.3) can be calculated based on the least square fit-based technique [92]. The corresponding Z_G equals to the dominant eigenvector of the complex sum of squares and products matrix (S_Z) as mentioned in Section 2.8.

$$S_Z = \sum_{i=1}^{N_g} (Z_i Z_i^*)/(Z_i^* Z_i) \tag{6.4}$$

6.3.3 PD-based gait similarity measurement and its constraint to walking speed change

Given two PMSs of two gaits (Z_{G_1} and Z_{G_2}), PD can measure the similarity which is invariant to translation, scaling and rotation as mentioned in Equation (2.33), where $d_P \in [0,1]$. The smaller value of $d_P(Z_{G_1}, Z_{G_2})$, the more possibility that gait features Z_{G_1} and Z_{G_2} belong to the same subject.

In fact, boundary segments corresponding to the different body parts changes differently (i.e. regarding both manners of magnitude and direction) due to the change of walking speed. Thus, it will be inefficient to treat the gait shape as a whole where PD is applied for similarity measurement between the gaits from different walking speeds.

In this chapter, *DCM* is proposed to deal with this constraint. It improves the performance of cross-speed gait similarity measurement. *DCM* contains key parameters to decompose gait shape boundary into a few boundary segments in a way such that each segment contains boundary points that are similarly affected by walking speed change. Then, PD is applied on each boundary segment instead of the whole shape boundary.

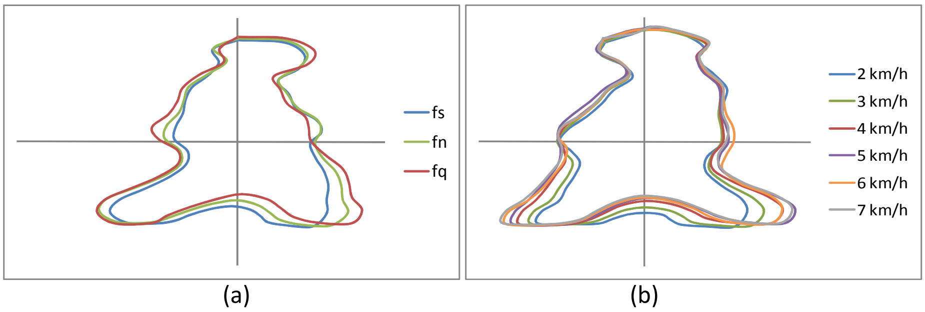


Figure 6.2: Examples of PMS-based gait feature from various speeds. (a) The CASIA gait database C: fs is slow walking, fn is normal walking and fq is fast walking. (b) The OU-ISIR gait database.

6.4 Impacts of Speed Change on PSA-based Gait Recognition

This section comprehensively analyses the impacts of walking speed change on the *PSA*-based gait recognition. It will reveal the following key points from different perspectives: 1) how significant the impacts are when the walking speed changes in various degrees? 2) how the individual body parts will be affected when the walking speed changes? 3) how the individual person responds to the change of walking speed differently?

Figure 6.2 shows examples of PMS-based gait feature under various walking speeds. It can be seen that arms swing higher, legs lift up higher and stride length is longer when walking speed is faster. To provide more detailed investigations on these basic observations, the experiments are set up in the following way. Six subjects from the OU-ISIR gait database are randomly selected and used in the analyses. Through the analyses, each shape boundary contains 40 re-sampled points. The first point is located at head position and the rest of the points are distributed along the shape boundary in a clockwise direction as illustrated in Figure 6.3 (a).

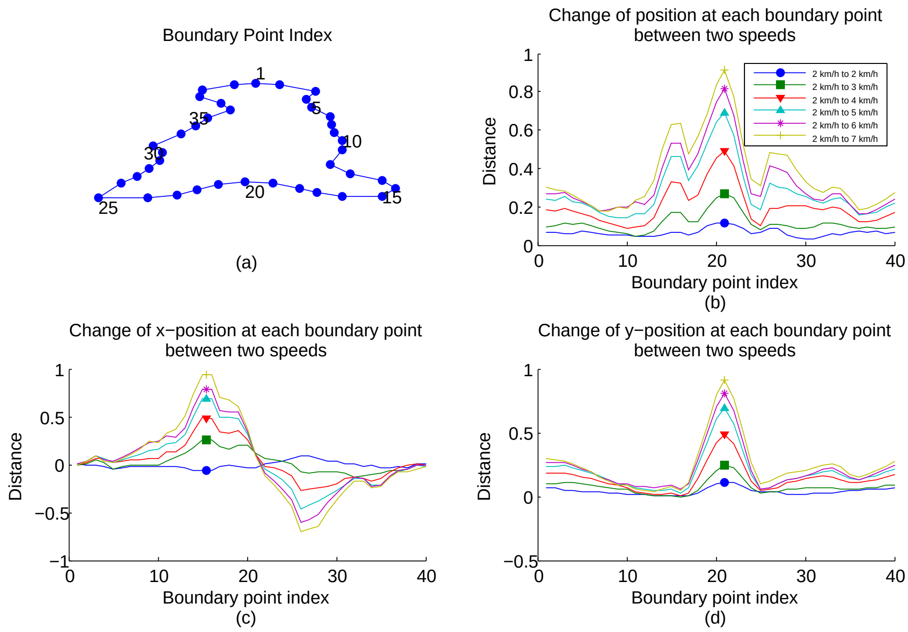


Figure 6.3: Boundary point variations between different speeds. (a) Boundary point index. (b) Change of position. (c) Change of x-position. (d) Change of y-position.

6.4.1 The change of walking speed vs. the change of walking pattern

In this *Analysis-A*, when the walking speed changes, the experiments record the traveling distance of each re-sampled boundary point on PMS from its original position under a previous walking speed to its new position under a new walking speed. The distance (d_E) is measured on the relevant Cartesian coordinates shown in Figure 6.3. If $P_{i,s_1} = (x_{i,s_1}, y_{i,s_1})$ and $P_{i,s_2} = (x_{i,s_2}, y_{i,s_2})$ are the i^{th} boundary points from two different walking speeds (s_1 and s_2) in Euclidean 2-space, then the distance (d_E) from P_{i,s_1} to P_{i,s_2} is given by Equation (6.5).

$$d_E(P_{i,s_1}, P_{i,s_2}) = \sqrt{(x_{i,s_1} - x_{i,s_2})^2 + (y_{i,s_1} - y_{i,s_2})^2} \tag{6.5}$$

In Figure 6.3, it can be clearly seen that large speed change has bigger impact than small speed change. For the case without speed change (2 km/h to 2

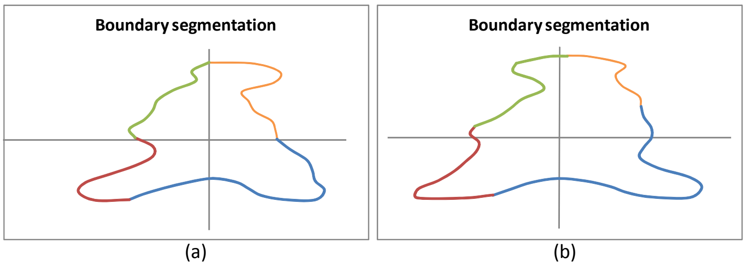


Figure 6.4: Boundary segmentation. (a) The CASIA gait database C. (b) The OU-ISIR gait database.

km/h), the small shape deviation is mainly caused by the minor inconsistency of individual gait. Even more importantly, Figure 6.3 (c) shows that the boundary points from index 10 to 21 (corresponding to back leg) shift to the right side and the boundary points from index 21 to 32 (corresponding to front leg) shift to the left side when speed increases. It is consistent with the observation in Figure 6.2 that stride length (index 10 to 32) is longer when walking speed is faster.

In Figure 6.3 (d), three major segments are observed to contain significant changes on y-axis component when walking speed changes. They are back upper body (index 1 to 11) which includes back arm, front upper body (index 30 to 40) which includes front arm, and legs apart (index 17 to 25). That is, arms and legs normally lift up higher when walking speed increases.

Based on this analysis, we can locate key boundary points to split the original boundary into a few segments. Inside each segment, the variation trends of the boundary points caused by walking speed change are similar. According to the boundary point variations in Figure 6.3 (b), we locate local saddle points which are the points of index 8, 18, 23, 31 and 34. Since the points of index 18 and 31 are too close to the points of index 23 and 34 respectively, without over-segmenting the boundary, the experiment keeps only the points of index 23 and 34. This is because they are more clear to indicate changes of the variation trends (see Figure 6.3 (b), (c) and (d)).

Therefore, the three reference saddle points (8, 23 and 34) are used to decompose the shape boundary into four segments as shown in Figure 6.4. Inside each segment, the variation trends of the boundary points caused by walking speed change are similar. In this chapter, this boundary segmentation will be further optimized and used in the proposed *DCM* (see Section 6.5.2) in order to improve cross-speed gait recognition performance.

6.4.2 The change of walking speed vs. the higher-order derivatives of the shape boundary

In this *Analysis-B*, the impacts caused by walking speed change on the higher-order derivatives of the shape boundary are further investigated. The first three higher-order derivatives of the shape boundary (i.e. tangent, curvature, and aberrancy) are briefed here, followed by the generic representation.

Given two boundary points $P_0 = (x_0, y_0)$ and $P_1 = (x_1, y_1)$, the first order derivative $f'(P_0, P_1)$ of a discrete function, tangent, using Forward Divided Difference [140] is given in Equation (6.6).

$$\begin{aligned} f'(P_0, P_1) &= \frac{y_1 - y_0}{x_1 - x_0} \\ &= \frac{y_0}{x_0 - x_1} + \frac{y_1}{x_1 - x_0} \end{aligned} \tag{6.6}$$

Given three boundary points $P_0 = (x_0, y_0)$, $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, the second order derivative $f''(P_0, P_1, P_2)$, curvature, is given in Equation (6.7).

$$\begin{aligned} f''(P_0, P_1, P_2) =& \frac{f'(P_1, P_2) - f'(P_0, P_1)}{x_2 - x_0} \\ =& \frac{y_0}{(x_0 - x_1) \cdot (x_0 - x_2)} + \\ & \frac{y_1}{(x_1 - x_0) \cdot (x_1 - x_2)} + \\ & \frac{y_2}{(x_2 - x_0) \cdot (x_2 - x_1)} \end{aligned} \tag{6.7}$$

In the same way, given four boundary points (x_0, y_0), (x_1, y_1), (x_2, y_2) and (x_3, y_3), the third order derivative $f'''(x_0, x_1, x_2, x_3)$, aberrancy, is given in Equa-

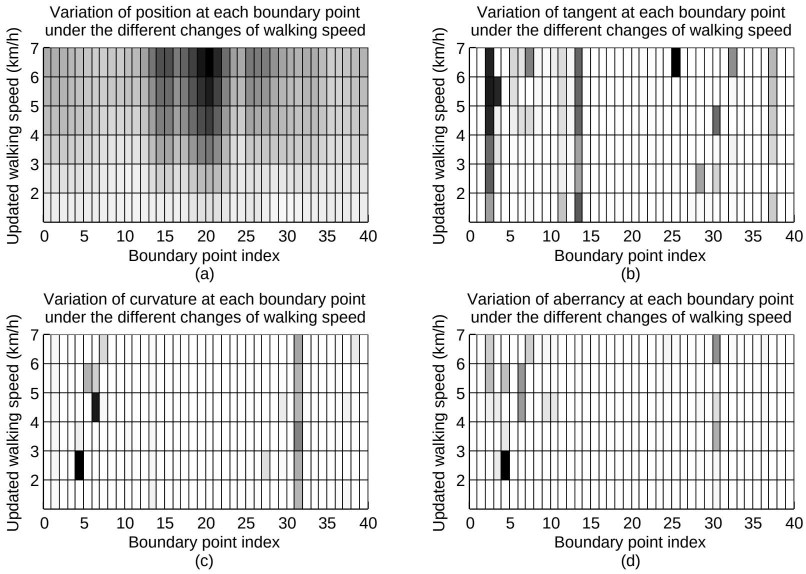


Figure 6.5: Efforts caused by the walking speed change on the variance of Euclidean distance, and its 1^{st} order (tangent), 2^{nd} order (curvature) and 3^{rd} order (aberrancy) derivatives respectively at each boundary point. The reference walking speed is 2 km/h. The updated walking speeds are shown on the corresponding y-axis. (a) Variation of position. (b) Variation of tangent. (c) Variation of curvature. (d) Variation of aberrancy. The darker color represents the higher impact.

tion (6.8).

$$\begin{aligned} f'''(x_0, x_1, x_2, x_3) =& \frac{f''(x_1, x_2, x_3) - f''(x_0, x_1, x_2)}{x_3 - x_0} \\ =& \frac{y_0}{(x_0 - x_1)\cdot(x_0 - x_2)\cdot(x_0 - x_3)} + \\ & \frac{y_1}{(x_1 - x_0)\cdot(x_1 - x_2)\cdot(x_1 - x_3)} + \\ & \frac{y_2}{(x_2 - x_0)\cdot(x_2 - x_1)\cdot(x_2 - x_3)} + \\ & \frac{y_3}{(x_3 - x_0)\cdot(x_3 - x_1)\cdot(x_3 - x_2)} \end{aligned} \tag{6.8}$$

For a more generic representation, higher-order derivatives $f^n(P_0, ..., P_n)$ of shape boundary can be calculated as follow.

$$f^n(P_0, ..., P_n) = \frac{f^{n-1}(P_1, ..., P_n) - f^{n-1}(P_0, ..., P_{n-1})}{x_n - x_0} = \sum_{i=0}^{n} \frac{y_i}{\prod_{j=0, j\neq i}^{n}(x_i - x_j)} \tag{6.9}$$

From Figure 6.5, it is seen that impacts caused by the speed change on tangent, curvature, and aberrancy (Figure 6.5 (b), (c), (d)) are much less than the impacts on global shape appearance (Figure 6.5 (a)). Particularly, from Figure 6.5 (c) and (d), it is seen that the speed change has no impact on legs motion (boundary point index 10 to 30) and only has minor impact on arms motion (boundary point index 4 to 8 and index 30 to 34). Therefore, the higher-order derivatives of shape information are more robust to the speed change than the global shape appearance.

The observations above motivate us to develop a new gait shape descriptor which is based on the higher-order derivatives of shape boundary information and is able to better tolerate the change of walking speed. The details can be referred to Section 6.5.1.

6.4.3 The change of walking speed vs. the impacts on the different body parts

From *Analysis-A*, it has indicated that different boundary segments corresponding to various body parts are affected differently when the walking speed changes. In this *Analysis-C*, a further investigation is conducted by measuring the difference quantitatively using PD on each boundary segment before-and-after the walking speed changes.

From Table 6.3, it can be seen that: 1) larger speed change causes larger shape change; 2) walking speed change has the different impacts on the different body parts; 3) the effect caused by the speed change on legs motion (segments 2 and 3: lower bodies) is much larger than the effect on arms motion (segments 1 and 4: upper bodies). This is because increasing in walking speed will increase

Table 6.3: Gait shape change (based on the measurement with PD) caused by speed change. Shape boundary is divided into four segments (see Figure 6.4) including segment1 (orange curve), segment2 (blue curve), segment3 (red curve) and segment4 (green curve).

Gallery	Probe	Segment1: back upper body	Segment2: back lower body	Segment3: front lower body	Segment4: front upper body
2 km/h	2 km/h	0.00011	0.00077	0.00072	0.00051
2 km/h	3 km/h	0.00018	0.00137	0.00204	0.00074
2 km/h	4 km/h	0.00038	0.00730	0.01009	0.00183
2 km/h	5 km/h	0.00040	0.01264	0.01709	0.00340
2 km/h	6 km/h	0.00041	0.01787	0.02137	0.00340
2 km/h	7 km/h	0.00044	0.02516	0.02846	0.00351

the stride length [14]. Naturally, arms will swing higher when the stride length increases.

So, it is better to decompose the original shape boundary into segments properly. Then, further processing is carried out on each segment. This will provide two major advantages for cross-speed gait similarity measurement. First, weights can be assigned to each segment. Larger weight is assigned to the segment which is less affected by the walking speed change and contains higher discriminability for gait recognition. Second, performance of the transformation involved in PD process will be improved by carried on the relevant operations on each segment instead of the whole shape boundary. This finding leads to the new *DCM* proposed in Section 6.5.2.

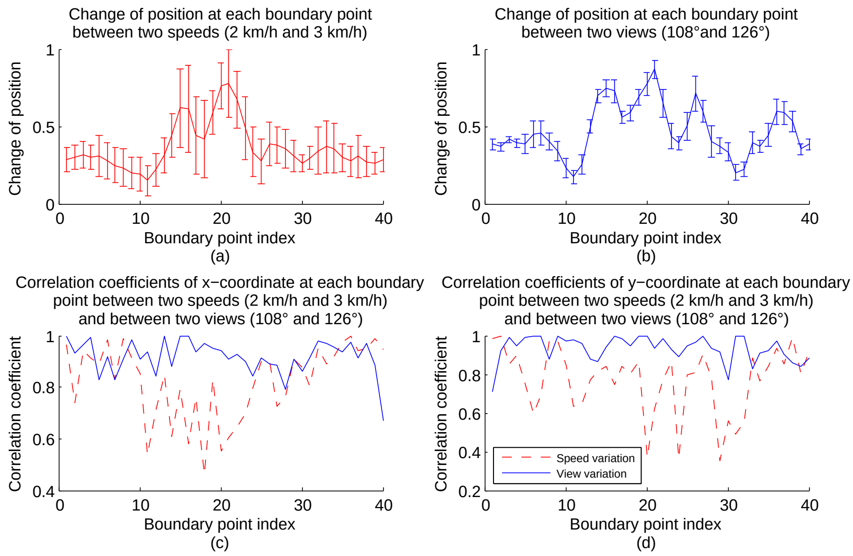


Figure 6.6: Patterns of shape change caused by speed change and view change. (a) Change of position caused by speed change. (b) Change of position caused by view change. (c) Correlations of x-coordinate between two different speeds and between two different views. (d) Correlations of y-coordinate between two different speeds and between two different views.

6.4.4 The change of walking speed vs. the response of the individual person

It is shown above that arms swing higher, legs lift up higher and stride length becomes longer when walking speed increases. However, each individual may respond differently to the walking speed change. This is not like the change of view which may cause the relatively same efforts on gaits of different persons [49][52]. In this *Analysis-D*, we will compare the effects on the individual gait caused by the changes of view and walking speed respectively.

Ten different persons are randomly selected and used in this analysis. The same experiments have been carried out by using more or less than ten randomly selected persons. The similar results were observed. The OU-ISIR gait database is used for analyzing the impacts caused by the speed variation and the CASIA gait database B is used for analyzing the impacts caused by the view variation.

Results are shown in Figure 6.6. Figure 6.6 (c) and (d) illustrate correlations of x- and y-coordinates respectively at each boundary point when walking speed changes or view changes. The correlation coefficient of the i^{th} boundary point (x- or y-coordinate) between two different walking conditions ($c1$ and $c2$) is given as following equation.

$$\rho_{P_{i,c1},P_{i,c2}} = \frac{E[(P_{i,c1} - \mu_{P_{i,c1}})(P_{i,c2} - \mu_{P_{i,c2}})]}{\sigma_{P_{i,c1}}\sigma_{P_{i,c2}}} \tag{6.10}$$

where $P_{i,c1} = (x_{i,c1}, y_{i,c1})$ and $P_{i,c2} = (x_{i,c2}, y_{i,c2})$ are the i^{th} point on gait shape boundary under walking condition $c1$ (e.g. walking speed is 2 km/h) and $c2$ (e.g. walking speed is 3 km/h) respectively, E is the expected value operator, $(\mu_{P_{i,c1}}, \sigma_{P_{i,c1}})$ and $(\mu_{P_{i,c2}}, \sigma_{P_{i,c2}})$ are expected values and standard deviations of $P_{i,c1}$ and $P_{i,c2}$ respectively. Equation (6.10) can be written as follow.

$$\rho_{P_{i,c1},P_{i,c2}} = \frac{\sum_{j=1}^{N_s}(P_{i,c1}^j - \overline{P}_{i,c1})(P_{i,c2}^j - \overline{P}_{i,c2})}{\sqrt{\sum_{j=1}^{N_s}(P_{i,c1}^j - \overline{P}_{i,c1})^2 \sum_{j=1}^{N_s}(P_{i,c2}^j - \overline{P}_{i,c2})^2}} \tag{6.11}$$

where $P_{i,c1}^j$ and $P_{i,c2}^j$ are the j^{th} sample of $P_{i,c1}$ and $P_{i,c2}$ respectively in the experimental dataset, $\overline{P}_{i,c1}$ and $\overline{P}_{i,c2}$ are the sample means of $P_{i,c1}$ and $P_{i,c2}$ respectively, and N_s is the total number of samples. The correlation coefficient ($\rho_{P_{i,c1},P_{c2}}$) shown in Figure 6.6 (c) and (d) is calculated as follow.

$$\rho_{P_{i,c1},P_{c2}} = \max_{P_{j,c2} \in S_i} |\rho_{P_{i,c1},P_{j,c2}}| \tag{6.12}$$

where S_i is a set of neighboring points of $P_{i,c2}$ in a shared local region.

In Figure 6.6, the pattern of shape change caused by walking speed change is much less generic than the pattern of shape change caused by view change. The intuitive explanation on the above observation is that view change is a kind of external dynamic factor which will affect the shooting conditions but is independent to subjects being shot. Thus, it will have the similar impacts to all subjects. However, walking speed change is up to every subject so it is a kind of internal factor strongly related to each individual person. Thus, it is unlikely to

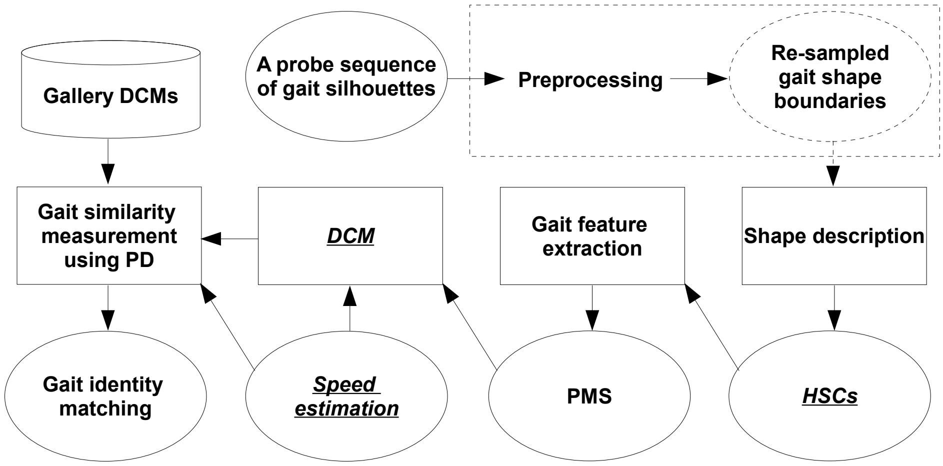


Figure 6.7: The speed-invariant *PSA*-based gait recognition.

use a generic transformation model [9][54] to tackle the speed change problem. Instead, this chapter will explore a solution to construct a new speed-invariant gait feature.

6.5 Speed-invariant Gait Recognition based on Improved PSA

According to the standard *PSA* framework for gait recognition in Section 6.3 and the comprehensive analyses on the impacts caused by walking speed change in Section 6.4, we propose a new *PSA*-based gait recognition which can tolerate the variation of walking speed. The framework of speed-invariant *PSA*-based gait recognition is shown in Figure 6.7. An algorithm pseudo-code is given in Algorithm 3. The pre-processing is referred to Figure 6.1 as discussed in Section 6.2. The proposed techniques to address the challenges of walking speed change are marked by italic and underline in Figure 6.7 and will be explained in the rest of this section.

Algorithm 3 The speed-invariant *PSA*-based gait recognition

Input: A probe gait sequence (G_0) and a set of gallery gait sequences ($\mathbb{G} = \{G_i\}_{i=1}^{N_s}$)
Output: Best identity matching of G_0 in $\mathbb{G}$

```
1: for i = 0 to N_s do
2:    Estimate the gait period of G_i
3:    Capture a set of gait shapes ({g_j}_{j=1}^{N_g}) from G_i within complete gait period(s)
4:    Estimate the gait speed (s_i) of G_i
5:    for j = 1 to N_g do
6:       Extract the shape boundary (b_j) from g_j
7:       Estimate head (H) and feet (F_L,F_R) positions of b_j
8:       Compute the re-sampled shape boundary (r_j) from b_j using H, F_L and F_R as reference
         points
9:       Generate the shape descriptor (Z_j) from r_j using the higher-order shape information
10:   end for
11:   S_Z ← Σ_{j=1}^{N_g} (Z_j Z_j^*)/(Z_j^* Z_j)
12:   Z_{G_i} ← dominant eigenvector of S_Z
13:   Generate Differential Composition Model (Γ_i) for Z_{G_i}
      Γ_i ← {P_k, w_k^{sc°}, Z_{G_i}^k}_{k=1}^{N_d}
14: end for
15: identity ← argmin_i Σ_{k=1}^{N_d} w_k^{sc°} d_p(Z_{G_0}^k, Z_{G_i}^k), i = 1, ..., N_s and sc° = |s_0 − s_i|
16: return identity
```

6.5.1 Shape descriptor based on HSC

This section is related to line 9 in Algorithm 3 pseudo-code. From *Analysis-B*, it is shown that walking speed change has no significant impacts on higher-order derivatives of shape boundary such as tangent, curvature, and aberrancy. Hence, *HSC* is proposed for describing gait shape. First, the higher-order derivatives of the shape boundary are calculated on the corresponding x- and y-components separately. Then, they are combined together to replace *CSC* inside the framework of *PSA*. HSC_r stands for r-order derivative of the shape boundary. HSC_r is a shape configuration (Z) which is written as follow.

$$\begin{aligned} HSC_r &= \left\{ \widetilde{0}, \frac{d^r z_i}{di^r} | i = 1, 2, ..., N_p \right\}^T \\ &= \left\{ \widetilde{0}, (f_x^r(i), f_y^r(i)) | i = 1, 2, ..., N_p \right\}^T \end{aligned} \tag{6.13}$$

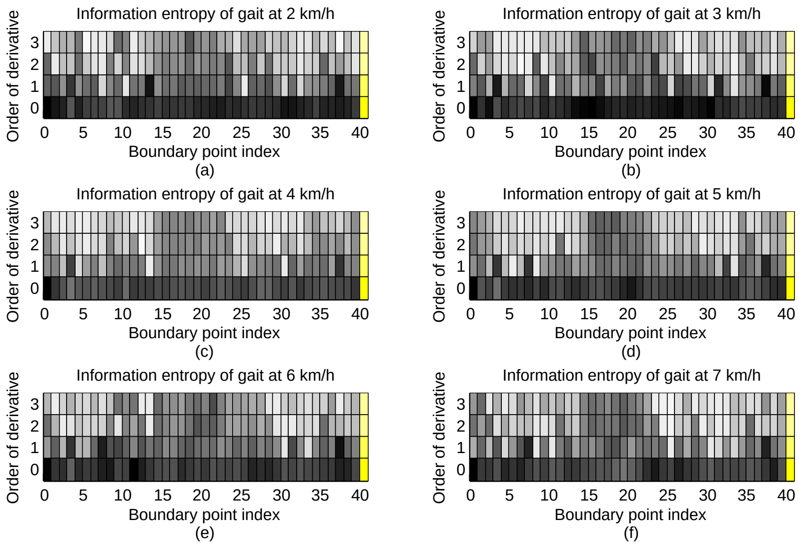


Figure 6.8: Information entropy representing quantity of gait information. (a)-(f) Walking speeds are 2,3,4,5,6,7 km/h respectively. The darker color represents the higher information entropy.

where $\widetilde{0}$ is (0,0) which is chosen as a reference, $z_i = (x_i, y_i)$ is the i^{th} boundary point, $\frac{d^r z_i}{di^r}$ is the r^{th}-order derivative of z_i on i, $f_x(i)$ is the x-component on boundary point i such that $f_x(i) = x_i$, $f_y(i)$ is the y-component on boundary point i such that $f_y(i) = y_i$. $f_x^r(i)$ and $f_y^r(i)$ are the r^{th}-order derivatives of $f_x(i)$ and $f_y(i)$ respectively. The derivative of a discrete function can be estimated using Forward Divided Difference [140]. In this particular case, the data points $f_x(i)$ and $f_y(i)$ are equidistantly distributed along the shape boundary. Then, general Forward Divided Difference can be calculated via Forward Difference which is defined as follow.

$$\begin{aligned} f_x^r(i) &= f_x^{r-1}(i+1) - f_x^{r-1}(i) \\ f_y^r(i) &= f_y^{r-1}(i+1) - f_y^{r-1}(i) \end{aligned} \tag{6.14}$$

For example, HSC_1 which stands for tangent geometrical information is calculated from the first derivative as follow.

$$\begin{aligned} f_x^1(i) &= f_x(i+1) - f_x(i) = x_{i+1} - x_i \\ f_y^1(i) &= f_y(i+1) - f_y(i) = y_{i+1} - y_i \end{aligned} \tag{6.15}$$

HSC_2 geometrically stands for curvature and is calculated from the second derivative as follow.

$$\begin{aligned} f_x^2(i) &= f_x^1(i+1) - f_x^1(i) = x_{i+2} - 2x_{i+1} + x_i \\ f_y^2(i) &= f_y^1(i+1) - f_y^1(i) = y_{i+2} - 2y_{i+1} + y_i \end{aligned} \tag{6.16}$$

In the same way, we can calculate HSC_3 which represents aberrancy from the third derivative as follow.

$$\begin{aligned} f_x^3(i) &= f_x^2(i+1) - f_x^2(i) = x_{i+3} - 3x_{i+2} + 3x_{i+1} - x_i \\ f_y^3(i) &= f_y^2(i+1) - f_y^2(i) = y_{i+3} - 3y_{i+2} + 3y_{i+1} - y_i \end{aligned} \tag{6.17}$$

In our experiments, we observe that an increasing in the order of derivative for HSC will be more robust to gait shape change caused by walking speed change (see *Analysis-B*). However, the higher order of derivative can capture less discriminative gait information. We adopt information entropy (H) [145] to measure a quantity of gait information from various orders of derivatives. The information entropy of the i^{th} boundary point (H_i) can explicitly be written as follow.

$$H_i = -\sum_{j=1}^{N_s} p(P_i^j)\ log\ p(P_i^j) \tag{6.18}$$

where p denotes the probability mass function, P_i^j is the j^{th} sample of the i^{th} boundary point along a given dataset for each order of derivative ($P_i = (f_x^r(i), f_y^r(i))$), and N_s is the total number of samples.

Figure 6.8 illustrates the information entropy of each boundary point from the different order derivatives on gait shape boundary. The shape boundary contains 40 points. The last column (yellow column) represents the average information entropy of the boundary points. The darker color represents the higher information entropy i.e. more gait information. Figure 6.8 shows that the higher order derivative of the shape boundary captures less gait information.

However, from Figure 6.10 which shows the discriminability (in terms of *Fisher discriminant ratio*) of different descriptions (in terms of different orders of derivative) of each boundary segment under various degrees of walking speed change, it can be seen that *HSC* has the higher discriminability for cross-speed gait recognition than *CSC*.

In practice, given the shape boundary is split into a few segments (see Section 6.5.2.1), a proper order of derivative ($\hat{r}$) for *HSC* can be determined under a degree of speed change (sc°) as follow.

$$\hat{r} = \underset{r}{\operatorname{argmax}} \sum_{i=1}^{N_d} w_{i,r}^{sc^{\circ}} \tag{6.19}$$

where $w_i^{sc^{\circ}}$ is the discriminability of the i^{th} boundary segment for gait recognition and sc° stands for the degree of speed change. In Section 6.5.2.2, we will propose a method for calculating $w_i^{sc^{\circ}}$. N_d is the total number of boundary segments and r is the order of *HSC*. According to our study, $r \in [0, 4]$ is sufficient (see Tables 6.4 and 6.5 for the detailed experimental results).

6.5.2 DCM

This section is related to lines 13 and 15 in Algorithm 3 pseudo-code. It has been shown in *Analysis-A* and *Analysis-C* that different boundary segments of gait shape are affected differently by speed change. It will make more senses that different boundary segments have to be treated differently and then integrated in a proper way when analyzing and recognizing gaits. Therefore, *DCM* is proposed to address this observed challenge. The model (Γ_i) representing the PMS-based gait feature (Z_{G_i}) is described in its general form as follow.

$$\Gamma_i = \{P_k, w_k^{sc^{\circ}}, Z_{G_i}^k\}_{k=1}^{N_d} \tag{6.20}$$

where P_k is the k^{th} decomposition boundary point (see *Analysis-A* and Section 6.5.2.1 for the details), $Z_{G_i}^k$ is the k^{th} clockwise boundary segment connecting P_k and P_{k+1} on Z_{G_i}, $w_k^{sc^{\circ}}$ is the discriminative weight assigned to $Z_{G_i}^k$, sc is the degree of speed change, $sc = |s_i - s_j|$ for any different values of s_j in the training dataset, s_i and s_j are walking speeds of G_i and G_j respectively, and N_d is the total number of decomposed segments. The calculations of these model parameters will be explained as follows.

When two gait features (Z_{G_1}, Z_{G_2}) are reconstructed in the forms of *DCM* (Γ_1, Γ_2), the overall dissimilarity D_P between the two gaits is calculated as a weighted sum of PDs as:

$$D_P(Z_{G_1}, Z_{G_2}) = \frac{\sum_{k=1}^{N_d} w_k^{sc^\circ} d_p(Z_{G_1}^k, Z_{G_2}^k)}{\sum_{k=1}^{N_d} w_k^{sc^\circ}} \tag{6.21}$$

6.5.2.1 Boundary decomposition

Based on the analysis in *Analysis-A* (see Figure 6.4), gait shape boundary is initially decomposed into four non-overlapped segments, namely segment1 (orange curve, back upper body), segment2 (blue curve, back lower body), segment3 (red curve, front lower body), and segment4 (green curve, front upper body) using the set of reference boundary points ($\mathbb{P} = \{P_k\}_{k=1}^{N_d}$) i.e. head position and the other three local minimums. According to *Analysis-A*, N_d and the initial value of $\mathbb{P}$ can be calculated. Given C_{P_j} standing for the movement of boundary point P_j caused by walking speed change (see Equation (6.5)), the optimal set of $\mathbb{P}$ is given as follow.

$$\hat{\mathbb{P}} = \underset{\mathbb{P}=\{P_k\}}{\operatorname{argmax}} \sum_{k=1}^{N_d} \Psi(C_{P_k}, C_{P_k+1}, C_{P_k+2}, ..., C_{P_{k+1}}) \tag{6.22}$$

where Ψ is the total correlation quantifying dependency among the set of variables i.e. $\mathbb{P}$ and P_k+1, P_k+2, ... are boundary points in clockwise direction along the boundary segment connecting P_k and P_{k+1}. Equation (6.22) can be regarded as an optimization process which adjusts boundary segments of cohesive boundary points. The total correlation can be calculated [145] as follow.

$$
\begin{aligned}
\Psi(C_{P_k}, ..., C_{P_{k+1}}) &= \sum_{j=0}^{P_{k+1}-P_k} H(C_{P_k+j}) - H(C_{P_k}, ..., C_{P_{k+1}}) \\
&= \sum_{c_{P_k} \in C_{P_k}} \cdots \sum_{c_{P_{k+1}} \in C_{P_{k+1}}} p(c_{P_k}, ..., c_{P_{k+1}}) log \frac{p(c_{P_k}, ..., c_{P_{k+1}})}{p(c_{P_k}), ..., p(c_{P_{k+1}})}
\end{aligned} \tag{6.23}
$$

where $H(C_{P_k+j})$ is the information entropy of variable C_{P_k+j} (see Equation (6.18)), $H(C_{P_k}, ..., C_{P_{k+1}})$ is the joint entropy of the variable set $\{C_{P_k}, ..., C_{P_{k+1}}\}$, $c_{P_k}, ..., c_{P_{k+1}}$ are training samples of $C_{P_k}, ..., C_{P_{k+1}}$ respectively, and p denotes the probability mass function.

6.5.2.2 Weight calculation

Sets of weights of discriminability ($\{w_k^{sc^\circ}\}_{k=1}^{N_d}$) are calculated for the different degrees of speed change (sc°). The weights are calculated based on *Fisher discriminant ratio* [146][147]. Given the training dataset U_T and the estimated walking speeds as follow.

$$
\begin{aligned}
U_{Ss,k}^{sc^\circ} &= \{(Z_p^k, Z_g^k) |\ Z_p, Z_g \in U_T,\ \ s(Z_p) = s(Z_g), |sp(Z_p) - sp(Z_g)| = sc\} \\
U_{Ds,k}^{sc^\circ} &= \{(Z_p^k, Z_g^k) |\ Z_p, Z_g \in U_T,\ \ s(Z_p) \neq s(Z_g), |sp(Z_p) - sp(Z_g)| = sc\}
\end{aligned} \tag{6.24}
$$

where $U_{Ss,k}^{sc^\circ}$ is the set of any two corresponding, k^{th}, boundary segments of the same subjects in U_T with the degree of speed change of sc°, $U_{Ds,k}^{sc^\circ}$ is the set of any two corresponding, k^{th}, boundary segments of different subjects in U_T with the degree of speed change of sc°, Z_p^k and Z_g^k are the k^{th} boundary segments of Z_p and Z_g, Z_p and Z_g are shape configurations of any two gaits in U_T, $s(Z_p)$ and $sp(Z_p)$ are the subject ID and the walking speed of Z_p respectively. To obtain *Fisher discriminant ratio*, the relevant means and variances of $U_{Ss,k}^{sc^\circ}$ are calculated as follow.

$$
\mu_{Ss,k}^{sc^\circ} = \frac{1}{N_{Ss,k}^{sc^\circ}} \sum_{(Z_p^k, Z_g^k) \in U_{Ss,k}^{sc^\circ}} d_p(Z_p^k, Z_q^k) \tag{6.25}
$$

$$(\sigma_{Ss,k}^{sc^\circ})^2 = \frac{1}{N_{Ss,k}^{sc^\circ}} \sum_{(Z_p^k, Z_g^k) \in U_{Ss,k}^{sc^\circ}} (d_p(Z_p^k, Z_q^k) - \mu_{Ss,k}^{sc^\circ})^2 \tag{6.26}$$

where $N_{Ss,k}^{sc^\circ}$ is the size of $U_{Ss,k}^{sc^\circ}$. Mean ($\mu_{Ds,k}^{sc^\circ}$) and variance ($(\sigma_{Ds,k}^{sc^\circ})^2$) of $U_{Ds,k}^{sc^\circ}$ are calculated in the same way as for $\mu_{Ss,k}^{sc^\circ}$ and $(\sigma_{Ss,k}^{sc^\circ})^2$. Then, inter-class (Between class) variance ($(\sigma_{B,k}^{sc^\circ})^2$), intra-class (Within class) variance ($(\sigma_{W,k}^{sc^\circ})^2$) and total variance ($(\sigma_{T,k}^{sc^\circ})^2$) are calculated as follow.

$$(\sigma_{B,k}^{sc^\circ})^2 = I_{Ss,k}^{sc^\circ} I_{Ds,k}^{sc^\circ} (\mu_{Ss,k}^{sc^\circ} - \mu_{Ds,k}^{sc^\circ})^2 \tag{6.27}$$

$$(\sigma_{W,k}^{sc^\circ})^2 = I_{Ss,k}^{sc^\circ} (\sigma_{Ss,k}^{sc^\circ})^2 + I_{Ds,k}^{sc^\circ} (\sigma_{Ds,k}^{sc^\circ})^2 \tag{6.28}$$

$$(\sigma_{T,k}^{sc^\circ})^2 = (\sigma_{B,k}^{sc^\circ})^2 + (\sigma_{W,k}^{sc^\circ})^2 \tag{6.29}$$

$$I_{Ss,k}^{sc^\circ} = \frac{N_{Ss,k}^{sc^\circ}}{N_{Ss,k}^{sc^\circ} + N_{Ds,k}^{sc^\circ}}, \; I_{Ds,k}^{sc^\circ} = 1 - I_{Ss,k}^{sc^\circ} \tag{6.30}$$

Hence, the weight value ($w_k^{sc^\circ}$) is calculated as follow.

$$w_k^{sc^\circ} = \frac{(\sigma_{B,k}^{sc^\circ})^2}{(\sigma_{T,k}^{sc^\circ})^2} \tag{6.31}$$

The analyses of the weights ($w_k^{sc^\circ}$) on each boundary segment k using different types of shape descriptor (CSC, HSC_r) under various degrees of speed change (sc°) are shown in Figures 6.9 and 6.10. Six subjects from the OU-ISIR gait database are used in these analyses.

Refer to CSC-graph in Figure 6.9, it can be seen that the discriminative capability of each boundary segment decreases when the degree of speed change

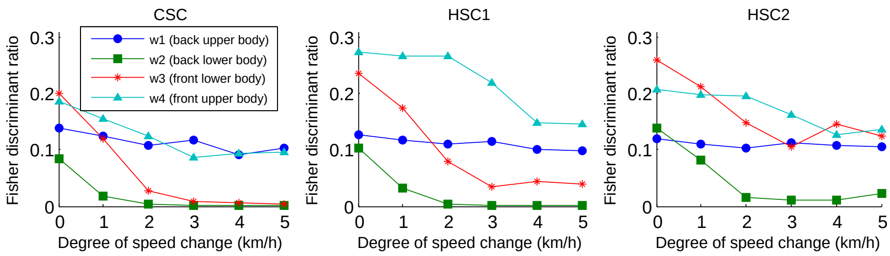


Figure 6.9: The weight analyses using different type of shape descriptors.

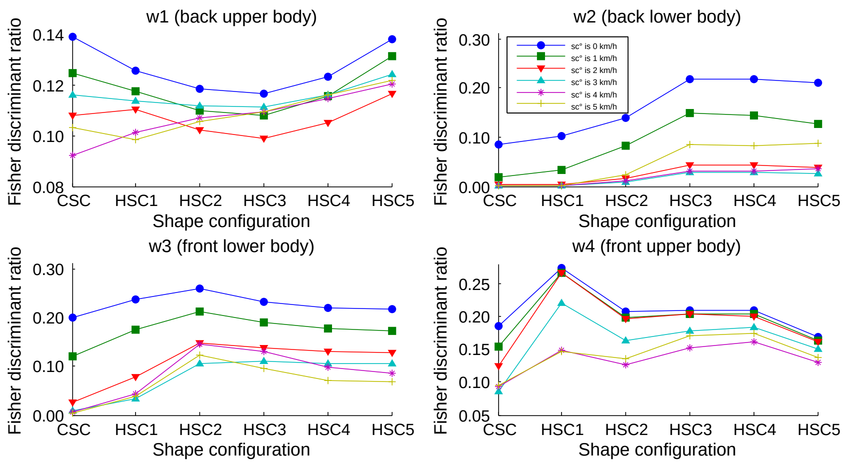


Figure 6.10: The weight analyses on different boundary segments corresponding to each body part.

increases. That is, larger speed change has higher impact on gait recognition. Besides, walking speed change affects legs' motion (lower bodies) much more than arms' motion (upper bodies). According to Figure 6.9, we may obtain more weight values for more cases of speed change by a proper interpolation. Examples of interpolated graphs for CSC, HSC_1 and HSC_2 are shown. Consequently, it can adapt more situations in testing process.

Refer to Figure 6.10, it is shown that HSC can efficiently increase the discriminative capability for every boundary segments, when compared with CSC. This is because HSC can describe gait shape boundary in the way that is more robust to gait shape change caused by walking speed change. When the order of HSC increases, the discriminability rises until it reaches the highest value before gradually dropping down. The reason is that *HSC* can tolerate the shape change caused by walking speed change but on the other hand undesirably captures less gait information. In our study, we recommend to determine the order of *HSC* according to Equation (6.19).

6.6 Experiments

In our experiments, four widely adopted gait databases are used to evaluate the performance of the proposed method, which include 1) the CASIA gait database C; 2) the OU-ISIR gait database; 3) the CMU Mobo gait database; and 4) the USF gait database. The first three databases directly support the study of gait recognition with respect to the variation of walking speed. The last database is considered as a real scene with non-controlled walking speed. Moreover, from the research perspective, there are different advantages from the first three databases: 1) the CASIA gait database C contains a large number of subjects; 2) the OU-ISIR gait database includes a large range of walking speeds; 3) the CMU Mobo gait database has been widely used by a large number of papers which can be referred for the comprehensive comparison. In our experiments, Nearest Neighbor (NN) [106] is used as a classifier.

In our experiments, the algorithm proposed in this chapter is implemented using the library functions of OpenCV 2.0 in Microsoft Visual C++ 8.0 environment on the computer with Quad Processor 2.66 GHz and 4 GB Ram.

Regarding the computational complexity, the training process may take time but the recognition process itself is fast given the trained models. There are three main training components: 1) determination of the order of *HSC* for each degree of speed change (see Equation (6.19)) takes about 5 minutes; 2) segmentation of gait shape boundary in *DCM* (see Equation (6.22)) takes about 10 minutes; 3) weight calculation in *DCM* for each degree of speed change (see Equation (6.31)) takes about 5 minutes. For example, total training time for the CASIA gait database C with 33 training subjects and 3 different walking speeds is approximately 40 minutes. The training process can be done offline beforehand.

In the recognition phase, given a probe sequence of gait silhouettes and the trained models, identity matching can be computed in less than 0.1 second. Com-

Table 6.4: Gait recognition performance (%) on the CASIA gait database C using the proposed method based on different types of shape configuration and PD without *DCM*.

Gallery	Probe	CSC	HSC_1	HSC_2	HSC_3	HSC_4
fs	fs	85	**95**	93	93	91
fn	fn	91	**96**	95	94	92
fq	fq	87	90	**91**	88	83
fn	fs	53	82	**87**	83	82
fs	fn	50	79	**85**	80	80
fn	fq	65	88	**89**	85	83
fq	fn	59	83	**86**	82	80
fs	fq	33	70	78	**86**	80
fq	fs	30	69	80	**84**	82

pared with the standard *PSA* framework for gait recognition (see Figure 6.1), the improved *PSA* framework for speed-invariant gait recognition (see Figure 6.7 i.e. *HSC* in place of *CSC* and adding *DCM* on top of PMS) takes approximately the same amount of time for the recognition process.

6.6.1 Results on the CASIA gait database C

The database contains 3 different walking speeds, namely slow (fs), normal (fn) and fast (fq). A total of 153 subjects are used in this experiment. 33 subjects are randomly selected and used in the training process, and the rest of 120 subjects are used to evaluate gait recognition performance. Refer to Section 6.2, fs, fn and fq are estimated as 4, 5 and 6 km/h respectively.

Tables 6.4 and 6.5 show gait recognition performance on the CASIA gait database C using the proposed method based on the different types of shape configuration. The results in Tables 6.4 and 6.5 are calculated using PD as simi-

Table 6.5: Gait recognition performance (%) on the CASIA gait database C using the proposed method based on different types of shape configuration and PD with *DCM*.

Gallery	Probe	CSC	HSC_1	HSC_2	HSC_3	HSC_4
fs	fs	86	**96**	93	93	92
fn	fn	93	**97**	96	95	94
fq	fq	88	91	**95**	92	86
fn	fs	62	86	**92**	87	87
fs	fn	60	81	**88**	83	80
fn	fq	74	89	**93**	90	85
fq	fn	68	85	**89**	85	82
fs	fq	52	77	83	**89**	86
fq	fs	53	79	83	**88**	87

larity measurement without and with *DCM* respectively. For the gait recognition under the same walking speed (see the first three rows in Tables 6.4 and 6.5), HSC_1 can achieve the most reliable performance. It works well to handle shape deviation caused by inconsistency of individual walking pattern. For the case of fast walking speed, HSC_2 achieves the best performance. This is because the consistency among individual gaits is more unstable when the individual walks faster. For the cross-speed recognition (see the last six rows in Tables 6.4 and 6.5), *CSC* absolutely fails but *HSC* still maintains high performance.

In our experiments, we may see that HSC_2 is sufficient when speed change is not significant e.g. ± 1 km/h (see rows 4-7 in Tables 6.4 and 6.5). For the case involving larger speed change i.e. ± 2 km/h (see the last two rows in Tables 6.4 and 6.5), HSC_3 achieves better performance. *HSC* successfully reduces impacts caused by speed change but also undesirably maintains less discriminative gait information which may have negative efforts on gait recognition performance. Thus, the order of derivative (r) for HSC_r has to be properly determined based

Table 6.6: Comparison of cross-speed gait recognition performance (%) on the CASIA gait database C using different methods.

(Gallery , Probe)	(fn , fn)	(fn , fs)	(fn , fq)	(fs , fq)
Uniprojective [30]	97	84	88	-
WBP [53]	**99**	86	90	60
GEI + 2DLPP [34]	65	62	63	-
EGEI + 2DLPP [34]	65	63	68	-
AEI + 2DLPP [34]	89	89	90	-
FDEI + 2DLDA [34]	88	89	90	-
HTI [56]	94	85	88	-
Pseudoshape [148]	98	82	92	-
Wavelet packet [112]	93	83	85	-
Orthogonal projections [113]	98	80	80	-
NDDP [114]	97	85	74	-
Gait curves [115]	91	65	70	-
The proposed method	97	**92**	**93**	**89**

on Equation (6.19). Moreover, the proposed *DCM* successfully improves the performance as seen in Table 6.5 when compared with the results in Table 6.4.

When compared with other existing methods which use the same database (see Table 6.6), the proposed method achieves comparable performance when gallery and probe walking speeds are the same ((fn,fn)) but better performance for cross-speed recognition ((fn,fs), (fn,fq), (fs,fq)) significantly for the case of large speed change ((fs,fq)).

Table 6.7: Gait recognition performance (%) on the OU-ISIR gait database using the proposed method without *DCM*.

Gallery(km/h) \ Probe(km/h)	2	3	4	5	6	7
2	100	96	84	72	72	72
3	100	100	96	80	76	60
4	76	96	96	92	92	80
5	76	76	96	96	100	96
6	64	68	80	96	100	96
7	56	68	80	96	100	100

6.6.2 Results on the OU-ISIR gait database

The database contains 6 different walking speeds from 2 km/h to 7 km/h with 1 km/h interval. Total 31 subjects are used in this experiment. Six subjects are randomly selected and used in the training process and the rest of 25 subjects are used to evaluate gait recognition performance.

The advantage of this database is that it includes a large range of walking speeds. The two benchmark methods [9][54] that directly address the challenges of walking speed change will be compared in this experiment. However, [54] uses a non-published gait database. The OU-ISIR gait database is the published database that can closely cover the similar scenarios of walking speed change as adopted in [54]. Thus, [54] is used only in a rough comparison and to indicate the trend of cross-speed gait recognition performance under different cases of walking speed change. The more accurate comparisons can be obtained in the other experimental sections of this chapter based on the other 3 gait databases.

Gait recognition performance on this database is shown in Tables 6.7 and 6.8. HSC_1 is used for gait recognition under the same walking speed. HSC_2 and HSC_3 are used for cross-speed recognition when speed change is up to 2 km/h and 4 km/h respectively. For other cases of large speed change, HSC_4 is used. The proposed method achieves average accuracies of 99%, 98%, 93%, 85%, 83%

Table 6.8: Gait recognition performance (%) on the OU-ISIR gait database using the proposed method with *DCM*.

Gallery(km/h) \ Probe(km/h)	2	3	4	5	6	7
2	100	100	88	80	80	84
3	100	100	100	88	84	80
4	88	96	100	92	92	84
5	96	96	96	96	100	96
6	84	84	96	96	100	100
7	84	88	84	96	100	100

and 84% for gait recognitions when differences between probe and gallery speeds are 0, 1, 2, 3, 4 and 5 km/h respectively.

The results in Tables 6.7 and 6.8 are compared as shown in Table 6.9. *DCM* significantly improves gait recognition performance especially for large speed change. This is because walking speed change has different impacts on each body part as shown in *Analysis-C*.

Table 6.10 shows the comparison between the proposed method (by using *DCM*) and the relevant published methods which have been investigated under the same/similar scenarios. The method [9] and the proposed method use the same OU-ISIR gait database of 25 subjects under the small speed change of ± 1 km/h (between 3 km/h and 4 km/h) and under the large speed change of ± 4 km/h (between 2 km/h and 6 km/h). In the experiments (as shown in Table 6.10), the dataset of slower walkers is used in Set A as the gallery, and vice versa in Set B. In order to provide more reference information, the experimental results of [54] are also included here in the comparison. Although [54] uses a non-published gait database (of 24 subjects) which is different from the OU-ISIR gait database, the reported speed change of ± 3.3 km/h (between 2.5 km/h and 5.8 km/h) is close (but less significant) to our case of large speed change.

From Table 6.10, when compared with [9] which uses the same OU-ISIR gait database, the proposed method performs much better for both cases of small

Table 6.9: Comparison of cross-speed gait recognition performance (%) on the OU-ISIR gait database using the proposed method between with and without *DCM*.

Degree of speed change (km/h)	Shape configuration	Without *DCM* (%)	With *DCM* (%)	Improvement (%)
0	HSC_1	98.67	99.33	1
1	HSC_2	96.80	98.00	1
2	HSC_2	85.00	92.50	8
3	HSC_3	75.33	85.33	10
4	HSC_3	66.00	83.00	17
5	HSC_4	64.00	84.00	20

and large speed changes. When compared with [54], it can be indicated that the proposed method can well tolerate the large speed change.

6.6.3 Results on the CMU Mobo gait database

The database contains 2 different walking speeds, namely slow walking (3.3 km/h) and fast walking (4.5 km/h). A total of 25 subjects are used in this experiment to evaluate gait recognition performance. The trained models are obtained from the training process based on the OU-ISIR gait database in Section 6.6.2. Therefore, the generic training process across different databases can be evaluated in this experiment. The proposed method is adopted with *DCM*. HSC_1 is used for gait recognition under the same walking speed while HSC_2 is used for cross-speed gait recognition.

From Table 6.11, when compared with other benchmark methods, it can be seen that the proposed method achieves comparable performance when gallery and probe walking speeds are the same. When the walking speed changes, the proposed method demonstrates very promising performance.

Table 6.10: Comparison of cross-speed gait recognition performance (%) using different methods.

Speed scenario		Stride normalization [54]	Transformation model [9]	The proposed method
Small speed change	Set A	-	84	100
	Set B	-	96	96
Large speed change	Set A	38	64	80
	Set B	29	52	84

6.6.4 Results on the USF gait database

The proposed method is further evaluated using the USF gait database which is a real scene captured in an uncontrolled environment. The USF gait database contains a set of 12 challenge experiments which are designed to investigate the effect of five factors affecting the performance of gait recognition. The five factors do not include the change of walking speed. In this database, the walking speed is not specifically controlled. That is, a person walks freely in various walking speeds in a non-controlled environment.

As a case study of real scene, the experiment A from the USF gait database which includes view variation is adopted to verify the proposed method. In the experiment A, probe gaits are recorded from the left camera while gallery gaits are recorded from the right camera. The two cameras' lines of sight are verged at approximately 30 degrees. A total of 122 subjects are used in this experiment. 22 subjects are randomly selected and used in the training process and the rest of 100 subjects are used to evaluate gait recognition performance. HSC_1 is used in our experiment.

In this section, two different experiments with different data settings are demonstrated here. The first experiment only adopts gaits from gallery dataset (but not from probe dataset) of the experiment A, so the views of gaits are the same. A gait sequence of each subject is divided into 2 subsequences of complete *walking cycle*(s). One subsequence is used as a probe data while another is used

Table 6.11: Comparison of cross-speed gait recognition performance (%) on the CMU Mobo gait database using different methods.

(Gallery, Probe)	(slow, slow)	(fast, fast)	(fast, slow)	(slow, fast)
pHMM [28]	-	-	-	84
CMU [103]	100	-	-	76
MIT [104]	100	-	-	64
NDDP [114]	100	100	80	88
Eigen feature [149]	96	96	-	75
LDT [150]	-	-	80	80
UMD [151]	100	100	84	80
SSP [152]	100	100	32	54
FSVB [153]	100	100	80	82
HMM [154]	72	68	56	32
SC [155]	100	100	84	80
The proposed method	100	100	88	92

as a gallery data. Leave-one-out cross-validation is applied in this evaluation. The proposed method achieves approximately 92% recognition performance.

The second experiment directly adopts the Experiment A of USF gait database which includes both gallery and probe datasets. Thus, both factors of view change and non-controlled walking speed are considered.

Table 6.12 shows that the proposed method is still robust to view change with non-controlled walking speed. It performs better or comparable when compared with other existing methods evaluated on the same database. By taking into account the encouraging performances for the cases of more significant speed changes (see experimental results based on the other databases above), the overall performance of the proposed method is clearly superior. For example, from Table 6.12, [28] performs slightly better than the proposed method. However, when

Table 6.12: Comparison of gait recognition performance (%) on the experiment A of the USF gait database using different methods.

Baseline [2]	pHMM [28]	*GEI* [27]	MSCT+SST [130]
73	85	83	80
HMM [132]	Eigen feature [149]	CFET [129]	The proposed method
80	78	83	83

walking speed change is more significant (see Table 6.11), the proposed method performs better than [28].

6.7 Conclusion

This chapter has conducted a comprehensive study aiming to reveal the impacts that different walking speeds exert on gait recognition. Our study has considerably enriched the performance of speed-invariant gait recognition by proposing the method based on the improved *PSA*. That is, *HSC* has been proposed as a robust shape descriptor which has been shown to be invariant to walking speed change. In this way, PMS is extracted as a novel speed-invariant gait feature from a set of *HSC*s describing a sequence of gait shapes from complete walking period(s). Then, PD has been used to measure gait similarity between two PMSs of any two gaits.

To enhance the performance of the cross-speed gait similarity measurement, *DCM* has been proposed to model PMS as a set of non-overlapped boundary segments. Then, the final gait distance is calculated from a weighted (i.e. *Fisher discriminant ratio*) sum of PDs corresponding to each boundary segment. Based on the comprehensive experimental results, our proposed method has outperformed other existing methods in the literature including the state-of-the-art methods.

Chapter 7

View-invariant Gait Recognition based on Low-rank Textures Analysis

7.1 Introduction

This chapter proposes a new method in the second category of *cross-view gait recognition* (see Section 1.2.2.2), which has several advantages as below. Compared with the first category (see Section 1.2.2.1), the second category uses a simpler non-cooperative camera system. Compared with the third category (see Section 1.2.2.3), the second category does not rely on supervised training for view-normalization process. Thus, the normalization is not limited to the trained views but is applicable for any view.

Compared with the state-of-the-art [44] of the same category (i.e. the second category), the proposed method performs view-normalization on the gait silhouettes instead of the key joints (i.e. lower limbs' poses). This will have two main benefits: 1) it does not require tracking of the key joints which can be unstable on markerless motion estimation; 2) it contains more reliable gait information after the view-normalization process, which can lead to better recognition performance.

The proposed method contains two main stages as summarized in this paragraph. First, in the view-normalization stage, a new *Gait Texture Image (GTI)* is proposed, which represents an original gait information on a sequence of gait silhouettes under a certain view. *Transform Invariant Low-rank Textures (TILT)* (see Section 2.9) is applied on *GTI* in order to recover the optimized domain transformation which represents the geometric projection of the gait information from its original view onto the common canonical view. The obtained domain transformation is then applied to project the corresponding gait silhouettes (that are used to construct *GTI* in the first place) onto the canonical view. In this way, a sequence of gait silhouettes from any view can be normalized onto the common canonical view.

Second, in the recognition stage, *Procrustes Shape Analysis (PSA)* (see Section 2.8) is applied on a sequence of the view-normalized gait silhouettes to construct a novel view-invariant gait feature. Then, gait similarity measurement is carried out under the common canonical view.

7.1.1 Rationale of the proposed method

The proposed *GTI* works well with *TILT* for the view-normalization. This is because *GTI* of any individual always has a lowest rank when it is constructed from the canonical view. Thus, *TILT* is to seek the domain transformation which can project *GTI* obtained from a certain view onto the canonical view by minimizing the rank of the texture image despite gross sparse errors. In this chapter, the canonical view is approximately the side view which has been regarded as the best view for recognizing gaits [6][7].

Such recovered domain transformation contains geometric representation of the scene covering the sequence of gait silhouettes which is used to construct *GTI* beforehand. Thus, the domain transformation can be used to project all corresponding silhouettes from the original view onto the canonical view. These view-normalized silhouettes will be used further for view-invariant gait recognition.

There are two main components in *TILT*: domain transformation (as mentioned above) and sparse error matrix. In our study, the sparse error matrix is used to model the noises caused by occlusion, shadow, and silhouette segmentation errors. It is also used to model high-rank textures in *GTI* caused by the asymmetry of partial walking pattern in a complete *walking cycle*. In this way, the domain transformation can better deal with high-rank textures in *GTI* caused by the view factor rather than by the noises.

As explained above, the normalized *GTI* will not be used directly for recognizing gait, but to derive the domain transformation which is the core in our view-normalization process. This is because it is not robust to the missing (i.e. unseen) gait information caused by view changes. In fact, gaits from different views contain different seen/unseen human body parts, while *TILT* can only align seen information. That is, a quality of the normalized *GTI*s onto the canonical view is degraded due to these degeneracies and singularities of gait information. Moreover, without proper frame synchronization, silhouette registration, and time normalization, *GTI* and its normalized version through *TILT* cannot be used directly for gait recognition.

In this study, to better recognize gait under view change, a new gait description is proposed and applied on a sequence of gait silhouettes after it is transformed onto the canonical view by its corresponding domain transformation. The new gait description is based on *PSA*.

PSA is a proper choice for our method because it has been proved as a statistical shape analysis which can tolerate the change of orientation of an object. Particularly, the gait shape inconsistencies caused by view change can be addressed to some extent using the improved scheme of *PSA* which was proposed in our previous work [62]. However, it solved only the problem of small view change where major gait information is not affected. This is because there was no proper process to rectify gaits onto a common view. In this chapter, *PSA*-based gait description is carried out after the view change is rectified using the proposed *TILT*-based domain transformation on *GTI*. In this way, *PSA*-based gait descriptor will be more robust.

In addition, according to the previous study [6][7], the ideal view for gait recognition is the side view which is very similar to the canonical view focused in this study. When a gait is originally acquired under a view closer to the side view, the performance of the proposed method will be better. However, when an original view under which a gait is acquired is far from the side view, the performance will drop because of the significant differences between gaits under the two views (i.e. original view and canonical view) with large offsets. This chapter will discuss and provide a reference as to how view change from side view to oblique view will affect the proposed method, although it still performs better than other existing methods of the same category.

7.1.2 Summary of contributions and advantages

In summary, the main contribution of this chapter is to propose a new framework of view-invariant gait recognition. It includes the following aspects.

- Seeking gait information (i.e. *GTI*) that can be normalized across different views.
- Proposing a novel view-normalization process through domain transformation by *TILT* on *GTI*.
- Normalizing gait silhouettes based on corresponding domain transformation.
- Computing a novel view-invariant gait feature through an improved scheme of *PSA* on view-normalized gait silhouettes.

Moreover, the proposed method contains several advantages over the existing methods. They are summarized as below.

- It does not require supervised learning in the view-normalization process. Thus, it will not limit *cross-view gait recognition* only to the trained views.
- It does not require tracking of the body joints which may be sensitive on markerless motion estimation [32].
- It achieves the high and stable recognition performance under approximate side views even when view change is large. For example, in our experiment, it achieves more than 80% recognition rate when view change is $\pm 36°$ between 72° and 108° on the CASIA gait database B.

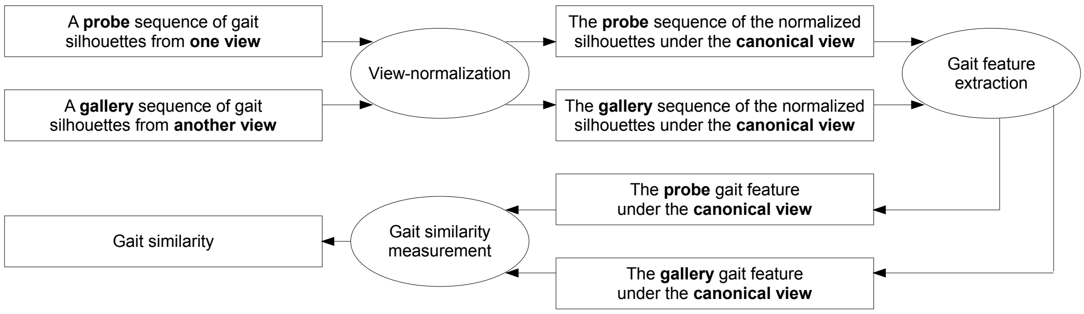


Figure 7.1: The proposed framework of view-invariant gait recognition.

- Unlike other methods especially in the third category, our view-normalization is performed in the input layer (i.e. gait silhouettes) instead of the feature layer (e.g. *Gait Energy Image (GEI)* [20], Gait Flow Image (GFI) [19]). In this way, we can freely select any proper techniques for gait feature extraction and similarity measurement.

- It uses a simple non-cooperative camera system.

The rest of this chapter is organized as follows. The framework is proposed in Section 7.2. View-normalization is explained in Section 7.3. Gait feature extraction and gait similarity measurement are discussed in Sections 7.4 and 7.5 respectively. Experimental results are shown in Section 7.6 and conclusions are drawn in Section 7.7.

7.2 Framework of the Proposed Solution

Figure 7.1 shows the framework of the proposed solution for view-invariant gait recognition (the detailed frameworks of its key processes will be illustrated in Figures 7.2 and 7.8). In these figures, rectangles represent inputs/outputs, while ellipses represent processing steps. Given a probe gait and a gallery gait recorded from different views, they are individually processed through the processes of view-normalization and feature extraction. Then, the similarity between the probe and gallery gait features is measured under a common canonical view.

As explained in Section 3.2, given a gait sequence from a video, a silhouette can be extracted from each frame using the method in [95]. However, some extracted silhouettes are incomplete. In this chapter, mathematical morphological operations [96] are used for holes remedy and noise elimination. Moreover, gait is analyzed within complete *walking cycle*(s) because it is a periodic action. The method proposed in Section 3.2 is adopted to estimate *gait period* of each gait sequence.

In the view-normalization process, *Gait Texture Image (GTI)* is extracted from a sequence of gait silhouettes within a complete *walking cycle*. It will be the input of low-rank texture optimization. *TILT* (see Section 2.9) is applied on *GTI* to seek a convex optimization that enables robust recovery of low-rank textures based on domain transformation despite gross sparse errors. In this way, *TILT* will transform *GTI* from any view into a common canonical view (i.e. approximate side view) where the low-rank textures are optimized. Another key component of *TILT* is sparse error matrix. It is used to eliminate errors/noises caused by corruption, occlusion, or shadow on gait image which may interfere the process of low-rank optimization. The recovered domain transformation is then re-applied to transform each corresponding gait silhouette into the canonical view. The sequence of view-normalized gait silhouettes will be further used in gait recognition procedure.

As mentioned in the introduction above, to address the challenge remaining from the view-normalization, a scheme of *PSA* is applied for gait feature extraction and similarity measurement as explained in Section 6.3. The pre-processes of shape boundary extraction and shape re-sampling are applied on each view-normalized gait silhouette to generate the re-sampled shape boundary which will be described using the proposed *Pairwise Shape Configuration (PSC)*. *PSC* describes a shape using a first-order derivative (i.e. tangent) of the shape boundary.

In *PSA*, Procrustes Mean Shape (PMS) is extracted from a set of *PSC*s in complete *walking cycle*(s) as a view-invariant gait feature. PMS is an average shape configuration computed from a given set of shape configurations (i.e. *PSC*s) by minimizing a sum of Euclidean distances between PMS and each configuration in the set. Then, the similarity between two PMSs of any two gaits from any two views is measured based on Procrustes Distance (PD) under the common canonical view. PD is calculated by minimizing Euclidean distance between two shape configurations regarding translation, rotation, and scaling.

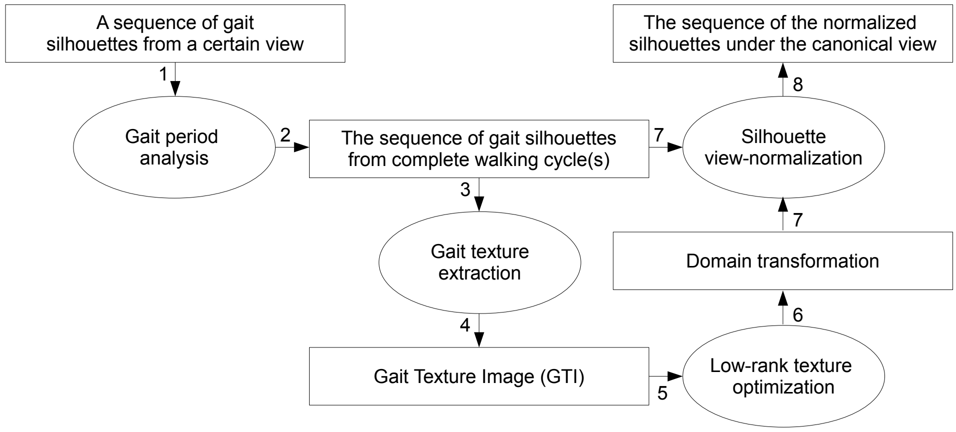


Figure 7.2: The proposed framework of view-normalization. The numbers present the orders of the processes.

7.3 View-normalization

Figure 7.2 shows the framework of the proposed view-normalization. The processes of *GTI* extraction and *TILT* for view-normalization are explained in this section.

7.3.1 Gait Texture Image

In this study, *GTI* is newly proposed to describe gait information based on the original gait silhouettes. *GTI* is not used as the final gait feature descriptor though. Instead, it is used as the input of *TILT* optimization process through which an optimized domain transformation will be obtained. This recovered domain transformation will be adopted and be applied on the original gait silhouettes to obtain the normalized silhouettes invariant to view changes.

Given a sequence of binary gait images $\{I_t(x,y)\}_{t=1}^{N_g}$ where $I_t(x,y)$ is a pixel at position (x,y) of gait image I_t and N_g is the number of gait images from complete *walking cycle*(s), *GTI* is obtained as follow.

$$GTI(x,y) = \frac{\sum_{t=1}^{N_g} I_t(x,y)}{N_g} \tag{7.1}$$

where the gait image is defined as a binary image which is obtained directly from foreground segmentation of each frame.

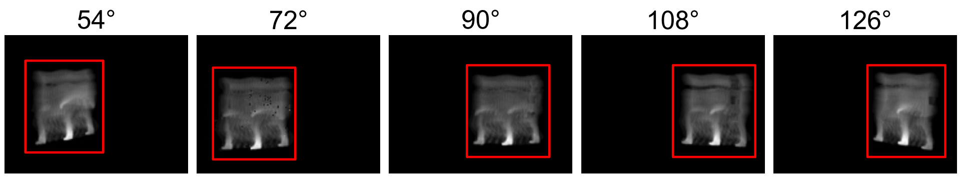


Figure 7.3: *GTI*s under various views. The red rectangles are the regions of interest in *GTI*s.

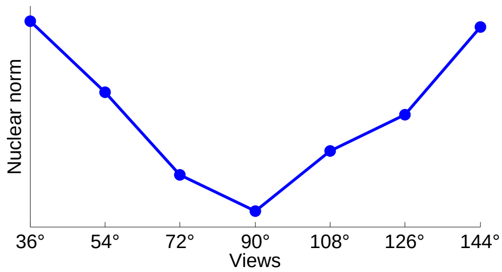


Figure 7.4: *Nuclear norm* of *GTI* from various views.

Figure 7.3 shows examples of extracted *GTI*s from various views on the CASIA gait database B. *TILT* will perform the optimization on the *region of interest (ROI)* in *GTI*. *ROI* is defined as the area just containing walking motion (see the red rectangles in Figure 7.3). It should not contain too much surrounding background which can confuse the low-rank optimization by *TILT*.

It has been indicated in the previous study [93] that *images of regular symmetric patterns always lead to low-rank textures*. From Figure 7.3, under side view (i.e. 90°), any gait shape itself is approximately symmetric. Moreover, gait shapes in the first half *walking cycle* roughly repeat the similar pattern of gait shapes in the second half *walking cycle*. Thus, gait shapes in a complete *walking cycle* also possess the (rough) symmetric property. In this way, *GTI* extracted from side view should contain low-rank textures when compared with other views.

To further confirm this observation, the *nuclear norm* is used to measure the rank of *GTI* from various views. The details of *nuclear norm* is explained in the optimization problem of *TILT* (see Section 2.9). Twenty four subjects from the CASIA gait database B are used in this analysis. From Figure 7.4, it can be confirmed that *GTI* from side view contains low-rank textures.

7.3.2 TILT for view-normalization

In this chapter, *TILT* (see Section 2.9) is applied for view-normalization of gaits. Based on our analysis in Section 7.3.1, GTI observed from a certain view can be regarded as a transformed version of the low-rank texture representing by GTI^0 from the canonical view (i.e. approximate side view). In addition, GTI can also be affected by the presence of partial corruption, occlusion, or shadow on gait images. All these factors can be modeled as sparse errors (E), which typically affect only a small fraction of *GTI*. In this way, the following equation is formulated.

$$GTI \circ \tau = GTI^0 + E \tag{7.2}$$

In this study, τ belongs to the homography group $GL(3)$ because it precisely describes a generic projective transformation in 2D. It can be used to approximate the changes of perceived positions of gaits when the observed view changes.

Thus, given a corrupted GTI observed from a certain view, *TILT* is applied on GTI to recover its low-rank texture GTI^0, the sparse error matrix E, and the domain transformation τ as the following optimization problem.

$$\min_{GTI^0, E, \tau} ||GTI^0||_* + \lambda ||E||_1 \qquad s.t. \;\; GTI \circ \tau = GTI^0 + E \tag{7.3}$$

where $||\cdot||_*$ denotes the *nuclear norm* or the sum of the singular values, $||\cdot||_1$ denotes the ℓ^1-norm or the sum of the absolute values of the entires, and GTI^0 is likely to represent gait from approximate side view. As mentioned in Section 2.9, the optimization problem in (7.3) is solved by the practical solution via successive convex programming [93].

The weighting parameter λ should be of the form $C/\sqrt{m}$ where C is a constant, typically set to unity [94]. In this study, m denotes the number of pixels of *ROI* in *GTI*. Moreover, λ can also be determined empirically. It can be adjusted by maximizing the recognition performance on the training dataset.

In the final stage, the computed domain transformation (τ) can well represent the geometric projection in the corresponding scene of *GTI*. That is, all gait

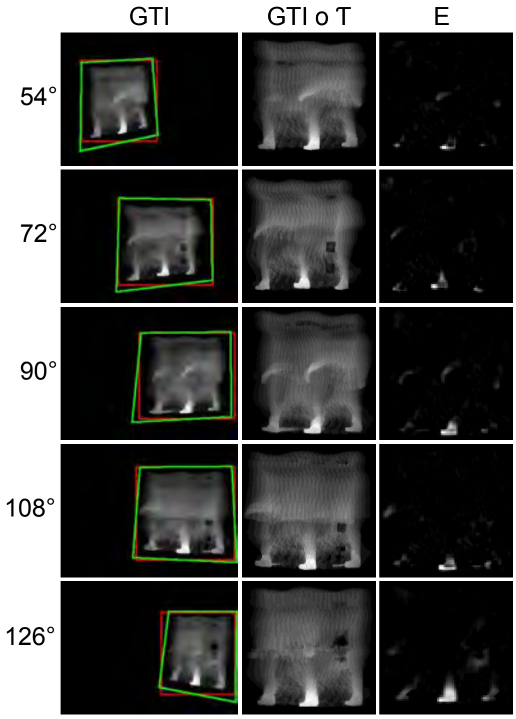


Figure 7.5: Sample results from *TILT* optimization. The first column is the input *GTI*. The red and green windows are the original *ROI* and the projected planar returned by *TILT*, respectively. The second column is the output $GTI \circ \tau$. The last column is the sparse error.

silhouettes embedded in GTI are also covered by the same geometric transformation. Thus, τ can be used to normalize the corresponding gait silhouettes (which was used to generate GTI) onto the canonical view as follow.

$$I_t^0 = I_t \circ \tau \tag{7.4}$$

Thus, $\{I_t^0\}_{t=1}^{N_g}$ is derived through (7.4) and will be used to generate view-invariant gait feature for view-invariant gait recognition.

Figure 7.5 shows the sample results from *TILT* optimization. It can be seen that GTI from any view can be well transformed into the common canonical view that is close to the side view. The sparse error matrix E can model the deviations caused by the corrupted pixels to some extent. In addition, it can also model some non-symmetric walking patterns in GTI from a complete *walking*

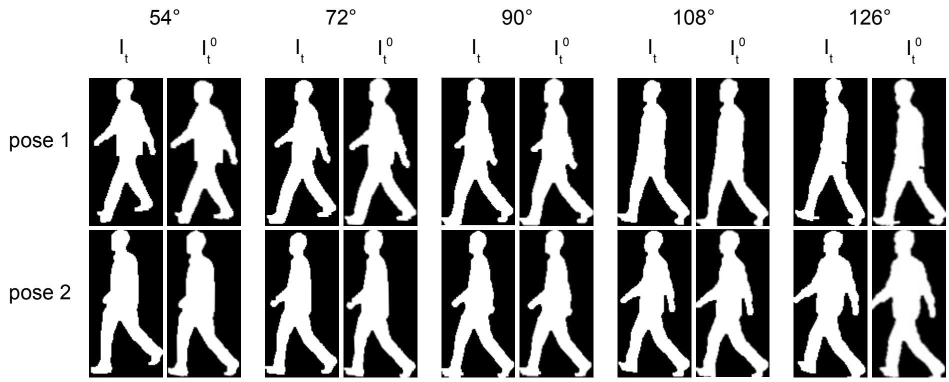


Figure 7.6: Sample transformed gait silhouettes from various views into the common canonical view. I_t is the original silhouette. I_t^0 is the transformed silhouette. Two gait poses are shown as examples. The transformation well rectifies the deformation on leg areas (e.g. walking trajectory) especially under 54° and 126°.

cycle. In this way, the domain transformation can better deal with the high-rank texture deformed by view factor.

Figure 7.6 shows the sample transformed gait silhouettes using the corresponding computed domain transformation. Particularly, under the views of 54° and 126°, it is clearly seen that the transformation well rectifies the deformation on leg areas.

To further evaluate the performance of the proposed view-normalization, we analyze the differences between various gaits on their original views and their transformed versions on the common canonical view after the transformation through (7.4). The gaits originally recorded from various views should be more close to each other on the canonical view after the view-normalization. To verify this assumption, we calculate the difference between the gait silhouettes by simple subtraction, where the positions of head [62] are used for silhouette alignment.

From Figure 7.7, the number of red and blue pixels (i.e. the differences) in the bottom row (i.e. after view-normalization) is less than the ones in the top row (i.e. before view-normalization). That is to say, the silhouettes from different views become closer after the view-normalization process. However, they are still dissimilar to some extent. This is because *TILT* can only normalize the visible information as mentioned before. Thus, there still exists the degeneracies and

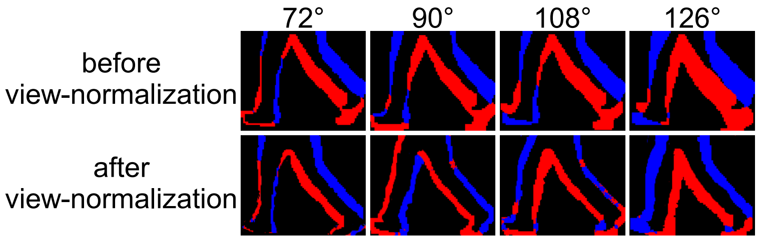


Figure 7.7: The difference between two silhouettes (focusing legs of pose 2 in Figure 7.6) from two different views (θ_1, θ_2). The reference view (θ_1) is 54°. Another view (θ_2) is listed in each column. The images in the first and second rows are calculated before and after view-normalization, respectively. Red pixels are gait information visible in θ_1 but not in θ_2. Blue pixels are gait information visible in θ_2 but not in θ_1. Black pixels are gait information visible in both θ_1 and θ_2.

singularities caused by invisible gait information especially in the case of large view difference (e.g. 54° versus 126°).

Considering the dissimilarity between gaits originally recorded from different views, it can be observed that the trend of the dissimilar gait parts on the canonical view is quite consistent. This is mainly because the domain transformation by *TILT* well rectifies the deformation of visible gait information, and invisible gait information usually wraps along the visible gait parts. As long as the visible parts can be well rectified, the boundaries of invisible gait parts will be also aligned. This will be further addressed in the process of gait feature extraction based on the statistical shape analysis (see Section 7.4).

7.4 Gait Feature Extraction

In this study, *PSA* scheme as explained in Sections 2.8 and 6.3 is adopted for gait feature extraction. As mentioned, *PSA* is a process of performing shape preserving Euclidean transformation on a set of shapes. It is able to achieve similarity measurement between two sets of shapes by properly superimposing. This property is useful for gait recognition because gait is a periodic dynamic action. During the superimposition, the positions and the sizes of gait shapes

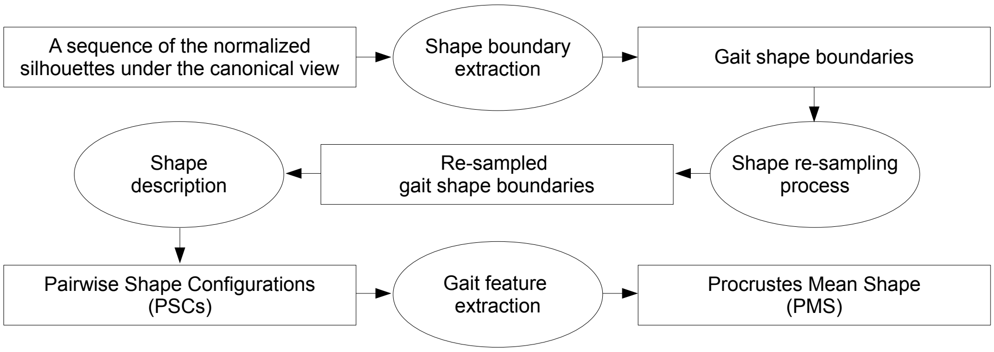


Figure 7.8: The proposed framework of gait feature extraction.

that vary throughout a *walking cycle* and a distance to camera are adjusted by proper translation, rotation and scaling.

Figure 7.8 shows the framework of the proposed gait feature extraction. Given the sequence of view-normalized gait silhouettes $\{I_t^0\}_{t=1}^{N_g}$ (see Section 7.3), it is used to construct view-invariant gait feature as explained in the rest of this section.

The pre-processing (e.g. shape boundary extraction and shape re-sampling process) of *PSA*-based gait recognition is discussed in Section 6.2. In this chapter, the number of re-sampled boundary points (N_p) is 100 where the boundary segments of head−right foot, right foot−left foot, and left foot−head contain 40, 20, and 40 points respectively.

7.4.1 Shape descriptor

In the conventional framework [90][106][134][144], the re-sampled shape boundary is described using *Centroid Shape Configuration (CSC)*. *CSC* is a global shape descriptor using the shape centroid as a global reference. The shape centroid is utilized as the origin of the 2-D shape space to register all shapes to a common center, which can handle translation invariance.

However, *CSC* has some disadvantages due to its global representation. In practice, gait shape of individual can be easily altered by many factors, particularly by the change of view and the inconsistency of walking pattern of the individual. Therefore, position of the shape centroid is not stable. Furthermore, according to our experiments, *CSC* which describes the shape based on its global shape appearance, is very sensitive to view change.

As demonstrated in Section 7.3, although the proposed view-normalization process can eliminate the significant shape inconsistencies caused by view change, the global gait shapes still vary across views due to different seen/unseen gait information recorded from different views. The proposed view-normalization through domain transformation by *TILT* well aligns the visible gait information (see Figure 7.7) which leads to the consistency of the local information (i.e. tangent) of gait shape boundaries across views (see Figure 7.10).

This challenge is addressed by replacing *CSC* with the proposed *Pairwise Shape Configuration (PSC)* which can better reveal the local shape information. This is because each boundary point is described by measuring the relation between the point itself and its local consecutive neighbor. By unwrapping the shape boundary into a set of boundary points, *PSC* can be described as a vector of complex numbers as follow.

$$Z = \left\{ \widetilde{0}, z_i \mid i = 1, 2, ..., N_p \right\}^T \tag{7.5}$$

where $z_i = (x_i - x_{i-1}) + j * (y_i - y_{i-1})$, (x_{i-1}, y_{i-1}) and (x_i, y_i) are the $(i-1)^{th}$ and i^{th} consecutive boundary points, $\widetilde{0}$ is (0,0), and N_p is the total number of boundary points. Since *PSC* encodes magnitude and direction of the tangent vectors, $\widetilde{0}$ is chosen as a reference frame (zero-degree tangent). $\widetilde{0}$ is placed at the starting point of *PSC*. The rest of boundary points are described relative to it.

7.4.2 PMS-based gait feature

As explained above, *PSC* provides a scheme to describe individual gait shape I_t^0 in each frame where I_t^0 is the normalized gait silhouette produced on the canonical view using domain transformation by *TILT*. Given the sequence of extracted *PSC*s $\{Z_t\}_{t=1}^{N_g}$ from $\{I_t^0\}_{t=1}^{N_g}$, Procrustes Mean Shape (PMS) is calculated as gait feature which integrates both shape and motion information.

In this chapter, Z_t represents *PSC* which is invariant to translation because each boundary point is described locally based on its neighboring boundary point. As mentioned in Section 2.8, given Z_t is invariant to translation, the corresponding

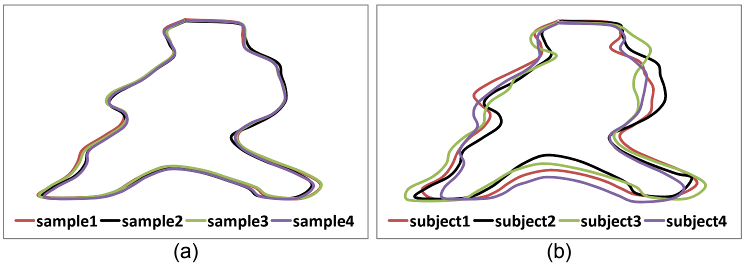


Figure 7.9: Example of PMS-based gait features on the CASIA gait database B. (a) Four samples of the same subject from the same view (90°). (b) Four different subjects from the same view (90°).

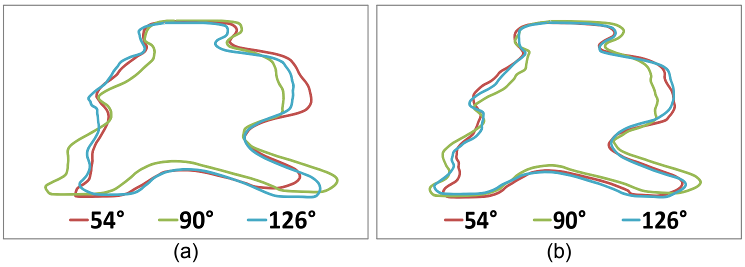


Figure 7.10: Example of PMS-based gait features of the same subject from three different views (54°, 90°, 126°) on the CASIA gait database B. (a) Without view-normalization process. (b) With view-normalization process.

PMS (Z_G) equals to the dominant eigenvector of the complex sum of squares and products matrix (S_Z).

$$S_Z = \sum_{t=1}^{N_g} (Z_t Z_t^*)/(Z_t^* Z_t) \tag{7.6}$$

In Figure 7.9 (a), it can be observed that PMS can efficiently preserve similarity between samples of the same subject from the same view. Figure 7.9 (a) also shows that small variation of PMS-based gait feature of an individual can occur due to small inconsistency of the individual walking pattern. In Figure 7.9 (b), it can be observed that PMS can efficiently preserve discriminative features between different subjects from the same view.

Figure 7.10 (a) shows that PMS-based gait features vary across views. In Figure 7.10 (b), it can be seen that PMS-based gait features from different views become more consistent when their corresponding gait silhouettes are aligned using the proposed view-normalization (through the domain transformation recovered in *TILT*) beforehand.

7.5 Gait Similarity Measurement

In the framework of *PSA* as explained in Section 2.8, PD is used to quantify the dissimilarity between two PMSs (Z_{G1} and Z_{G2}) of any two gaits as follow.

$$d_P(Z_{G1}, Z_{G2}) = 1 - \frac{|Z_{G1}^{*} Z_{G2}|^2}{||Z_{G1}||^2||Z_{G2}||^2} \tag{7.7}$$

where the superscript $*$ represents a complex conjugation transpose, and $d_P \in [0, 1]$. The smaller value of $d_P(Z_{G1}, Z_{G2})$, the more possibility that gait features Z_{G1} and Z_{G2} belong to the same subject.

7.5.1 Constraint to view change

Naturally, the different parts of gait will be affected differently when view changes. However, *PSA* performs proper rotation, translation, and scaling by considering each gait shape as a whole. Thus, for adjusting rotation in PD process to match PMS of gait from one view onto another view, the whole PMS will be properly rotated in one direction. This is not effective in practice as can be seen in Figure 7.11.

From Figure 7.11, the shape boundary is roughly divided into four segments using four reference points which are the boundary points at index 1, 20, 60 and 80 (P_1, P_{20}, P_{60}, P_{80}). These four points are manually detected regarding the changes of rotating direction for shape matching. It can be seen from Figure 7.11 that to match PMS of gait from one view (e.g. 90°) onto another view (e.g. 54°), the 1^{st} boundary segment (connecting P_1 and P_{20}) and the 3^{rd} boundary segment (connecting P_{60} and P_{80}) are rotated anticlockwise, while the 2^{nd} boundary segment (connecting P_{20} and P_{60}) and the 4^{th} boundary segment (connecting P_{80} and P_1) are rotated clockwise around the vertical axis. In this way, PD cannot take its full advantages of affine invariance if gait shape boundary is considered as a whole.

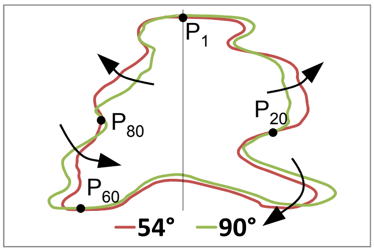


Figure 7.11: Rotation adjustment between PMS-based gait features of the same subject from two different views (54°, 90°). P_1, P_{20}, P_{60}, and P_{80} are the boundary points at index 1, 20, 60, and 80 respectively, which are used to decompose the shape boundary into 4 segments. The arrows indicate the directions of rotating the corresponding boundary segments from 90° to match the ones from 54°.

Thus, the gait shape boundary should be decomposed into a few boundary segments. Then, PD is applied on each boundary segment instead of the whole shape boundary. The key reference points can be optimized by maximizing the *cross-view gait recognition* performance on the training dataset based on various cases of view changes. All cases of view changes are likely to have the same shape boundary decomposition. In our experiment, the key reference points for boundary decomposition are the boundary points at index 1, 18, 58, and 84.

7.5.2 PD-based gait similarity measurement

Given that two gait features (Z_{G1} and Z_{G2}) are decomposed into segments based on the analysis above, the total dissimilarity D_P between the two gaits is calculated as a sum of PDs as follow.

$$D_P(Z_{G1}, Z_{G2}) = \sum_{k=1}^{N_d} d_P(Z_{G1}^k, Z_{G2}^k) \tag{7.8}$$

where Z_{G1}^k and Z_{G2}^k are the k^{th} boundary segments of Z_{G1} and Z_{G2} respectively, and N_d is the total number of boundary segments (i.e. $N_d = 4$ in this study).

7.6 Experiments

The CASIA gait database B is used in our experiments (see Sections 7.6.1, 7.6.2, and 7.6.3). It is a large multi-view gait database which contains 124 subjects. Five views ($54°, 72°, 90°, 108°, 126°$) are used to evaluate gait recognition performance under various views. In this chapter, other approximate frontal views are not considered in our experiments due to the fact that available visual features of gaits from the frontal views are too different from the canonical view. Therefore, it is inefficient to normalize them into the canonical view using the proposed view-normalization.

The comparisons on *cross-view gait recognition* within the same category (i.e. the second category) and across different category (i.e. the third category) are demonstrated in Sections 7.6.1 and 7.6.2 respectively. The proposed method is not compared with other methods in the first category because they are applicable for different scenarios. The proposed method (and all other methods in the second category) belongs to *cross-view gait recognition* where probe gait and gallery gait are recorded from two different views. On the other hand, methods in the first category belong to *multi-view gait recognition* where gallery gaits from multiple views are used to recognize probe gait from at least one view. Also, the robustness of the proposed method against increasing number of subjects in the dataset is demonstrated in Section 7.6.3

Moreover, our experiment in Section 7.6.4 is carried out based on a more practical database i.e. the USF gait database. It consists of persons walking in elliptical paths in front of cameras. This database is challenging because of several difficulties of the outdoor environment such as wind, shadow and illumination. It contains 122 subjects from two cameras (L and R). The cameras' lines of sight are verged at approximately 30 degrees.

In our experiments, Nearest Neighbor (NN) is used as a classifier for gait recognition.

7.6.1 Comparison with other methods using view-invariant gait feature (i.e. the second category)

The performance of *cross-view gait recognition* using the proposed method is evaluated and compared with other four methods in the second category (see Table 7.1) including: 1) *GEI* [27]; 2) view rectification using self-calibrating [44]; 3) *CSC+PSA* [106]; 4) *PSC+PSA* [62]. For a fair comparison, the same number of subjects (i.e. 65 subjects) are used by all methods in this experiment.

From Table 7.1, it is clearly seen that the proposed method significantly outperforms the baseline method [27] which simply matches *GEI* across views without any view-normalization, in all cases. When compared with the *PSA*-based methods [62][106], the proposed method achieves much better performance especially for the cases of large view changes. This is because the proposed view-normalization is applied to remove significant shape inconsistencies caused by view change before *PSA* is adopted and adapted properly in our method.

Moreover, the proposed method also performs better than the state-of-the-art of the second category [44] for most cases except a few cases related to 54° or 126°. According to our analysis, the method in [44] adopts the information of body joints in gait (which is not affected very much when view changes) so it is less sensitive to view change. However, it may also lose discriminability when removing such information highly sensitive to view change. The proposed method, instead, attempts to cope with as much information as possible from the original gait silhouette while eliminating the efforts caused by view change through *TILT* optimization process.

As shown in Table 7.1, the proposed method significantly outperforms the method in [44] for most cases including: 1) no view change (e.g. $\theta_p = 54°$ vs. $\theta_g = 54°$); 2) small view change (e.g. $\theta_p = 72°$ vs. $\theta_g = 90°$); 3) large view change under approximate side view (e.g. $\theta_p = 72°$ vs. $\theta_g = 108°$).

7.6.2 Comparison with other methods relying on learning mapping/projection relationship of gaits across views (i.e. the third category)

The performance of *cross-view gait recognition* using the proposed method is evaluated and compared with other four methods in the third category (see Table 7.2) including: 1) FT-*SVD* [52]; 2) *GEI-SVD* [50]; 3) GFI+*CCA* [48]; 4) *GEI+SVR* [49]. The methods in [49][50][52] use 100 subjects while the method in [48] uses 74 subjects from the CASIA gait database B for evaluating gait recognition performance. Based on our experiments, the proposed method is robust to increasing number of subjects. For a comprehensive comparison, it is evaluated using 100 subjects. Some missing results have not been reported by [48].

Obviously, the main advantage of the proposed method against the existing methods in this category is that it does not need any training process to build up the required mapping relations between different views. This is practically helpful in many cases because it cannot guarantee that all needed data will be available for training in the real world. Without the precise training process, the proposed method, as indicated in Table 7.2, still outperforms both *SVD*-based methods [50][52]. It also performs better than/comparable to the state-of-the-arts of the third category [48][49] in many cases.

7.6.3 Robustness when increasing a number of subjects in the dataset

This section is to evaluate robustness of our method when increasing number of subjects in the dataset. Usually, it is more difficult to recognize gaits when there are more subjects. Particularly, when view changes, gait of one subject from one view can be misidentified as gait of another subject from another view.

In this experiment, three cases of *cross-view gait recognition* using the proposed method are demonstrated including: 1) small view change (i.e. 72° vs. 90°); 2) large view change under approximate side walks (72° vs. 108°); 3) large view change under oblique and side walks (54° vs. 90°). Figure 7.12 shows the

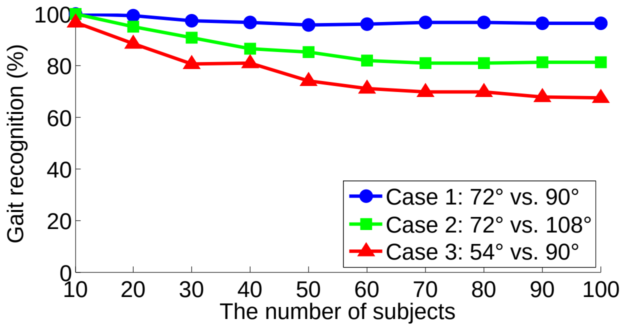


Figure 7.12: The performance of *cross-view gait recognition* using the proposed method against increasing the number of subjects in the dataset.

performances when increasing the number of subjects (10, 20, ..., 100 subjects) in the dataset.

It can be seen from Figure 7.12 that the performance of case 1 (72° vs. 90°) is very stable when the number of subjects increases from 10 to 100. This is because the proposed view-normalization performs well on the case of small view change. Moreover, the performances of cases 2 and 3 drop when the number of subjects increases from 10 to 50, but become stable from 60 to 100. Overall, the performance of the proposed method tends to be stable when increasing the number of subjects in the dataset. This will be helpful for practical uses.

7.6.4 Cross-view gait recognition under outdoor environment

The proposed method is further evaluated using the practical dataset (outdoor) i.e. the USF gait database. Among the 12 challenging experiments pre-defined for the USF gait database, the experiment A is adopted for our evaluation because it observes view change for gait recognition which matches the focus of this study.

Probe and gallery gaits are recorded from different cameras L and R respectively. From Table 7.3, the proposed method (which achieves 85%) perfroms better than/comparable to other methods in the literature.

7.7 Conclusion

This chapter has proposed a novel method for view-invariant gait recognition. *GTI* is first extracted from a sequence of gait silhouettes recorded under a certain view. *TILT* is then applied to seek a transformed version of *GTI* under the canonical view by minimizing the rank of the texture image based on domain transformation despite gross sparse errors. The recovered domain transformation is re-applied to normalize the corresponding gait silhouettes into the canonical view. Then, PMS is constructed as a view-invariant gait feature from *PSC*s describing the normalized gait silhouettes within a complete *walking cycle*. Gait similarity between two PMSs of any two gaits from any two views is measured based on PD under the common canonical view. Based on our promising recognition performance and several advantages over other existing methods, the proposed method is efficient for view-invariant gait recognition.

Our current experiments have been carried out based on widely adopted gait databases in which there are high quality gait silhouettes available. In the future work, we will investigate the robustness of the proposed method for view-invariant gait recognition in a cluttered environment where it has to deal with different noisy gait silhouettes. It can be seen in [93] that the attributes of low-rank optimization will be helpful for repairing such degraded dynamic texture by well adopting the correlation between frames in a motion sequence.

Table 7.1: Comparison on *cross-view gait recognition* (%) using different methods in the second category

Probe view (θ_p)	54°				
Gallery view (θ_g)	54°	72°	90°	108°	126°
GEI [27]	91	22	18	17	38
View rectification [44]	80	72	**74**	**70**	57
PSC+PSA [62]	**98**	63	50	20	33
CSC+PSA [106]	86	28	25	19	30
The proposed method	**98**	**79**	69	54	**59**
Probe view (θ_p)	72°				
Gallery view (θ_g)	54°	72°	90°	108°	126°
GEI [27]	23	94	50	34	21
View rectification [44]	65	85	73	59	**60**
PSC+PSA [62]	71	**98**	93	55	28
CSC+PSA [106]	33	88	69	55	26
The proposed method	**79**	**98**	**96**	**81**	57
Probe view (θ_p)	90°				
Gallery view (θ_g)	54°	72°	90°	108°	126°
GEI [27]	17	82	93	88	22
View rectification [44]	62	64	87	64	**70**
PSC+PSA [62]	52	93	97	77	28
CSC+PSA [106]	27	71	85	67	21
The proposed method	**69**	**97**	**98**	**93**	56
Probe view (θ_p)	108°				
Gallery view (θ_g)	54°	72°	90°	108°	126°
GEI [27]	17	36	69	94	33
View rectification [44]	**63**	70	69	82	68
PSC+PSA [62]	32	67	84	**97**	65
CSC+PSA [106]	24	54	68	80	27
The proposed method	49	**82**	**94**	**97**	**80**
Probe view (θ_p)	126°				
Gallery view (θ_g)	54°	72°	90°	108°	126°
GEI [27]	29	21	15	37	92
View rectification [44]	**63**	**67**	**67**	73	81
PSC+PSA [62]	28	29	32	67	**98**
CSC+PSA [106]	22	22	19	33	87
The proposed method	**63**	55	56	**80**	**98**

Table 7.2: Comparison on *cross-view gait recognition* (%) using different methods in the third category.

Probe view (θ_p)	54°				
Gallery view (θ_g)	54°	72°	90°	108°	126°
GFI+*CCA* [48]	–	88	60	**54**	47
SVR-rVTM [49]	91	**93**	59	51	42
GEI-dVTM [50]	91	81	49	31	27
FT-*dVTM* [52]	86	43	28	19	24
The proposed method	**98**	77	**68**	**54**	**56**
Probe view (θ_p)	72°				
Gallery view (θ_g)	54°	72°	90°	108°	126°
GFI+*CCA* [48]	–	–	–	–	–
SVR-rVTM [49]	**90**	94	91	67	44
GEI-dVTM [50]	85	94	84	52	31
FT-*dVTM* [52]	56	89	57	45	33
The proposed method	79	**98**	**96**	**81**	**54**
Probe view (θ_p)	90°				
Gallery view (θ_g)	54°	72°	90°	108°	126°
GFI+*CCA* [48]	60	96	–	**94**	**95**
SVR-rVTM [49]	63	92	93	91	65
GEI-dVTM [50]	52	75	93	79	45
FT-*dVTM* [52]	27	36	87	58	28
The proposed method	**70**	**97**	**98**	93	55
Probe view (θ_p)	108°				
Gallery view (θ_g)	54°	72°	90°	108°	126°
GFI+*CCA* [48]	**60**	**85**	**95**	–	**92**
SVR-rVTM [49]	42	77	**95**	94	**92**
GEI-dVTM [50]	38	57	91	94	84
FT-*dVTM* [52]	29	34	53	91	74
The proposed method	47	80	**95**	**97**	78
Probe view (θ_p)	126°				
Gallery view (θ_g)	54°	72°	90°	108°	126°
GFI+*CCA* [48]	53	**70**	75	94	–
SVR-rVTM [49]	42	57	**78**	**96**	92
GEI-dVTM [50]	31	42	53	80	92
FT-*dVTM* [52]	17	16	22	55	86
The proposed method	**58**	53	55	77	**98**

Table 7.3: Comparison on *cross-view gait recognition* (%) in the experiment A of the USF gait database using different methods.

Baseline [2]	pHMM [28]	*GEI* [27]
73	85	83
MSCT+SST [130]	HMM [132]	Eigen feature [149]
80	80	78
PEI+*LDA* [29]	CFET [129]	The proposed method
85	83	85

Chapter 8

Conclusions and Future Works

8.1 Summary of Research

In this thesis, changes of two key factors (i.e. an observed view and a walking speed) have been studied regarding their impacts on gait recognition. They are summarized as follow.

First, in practice, individual gaits can be observed from different views because: 1) a person can walk freely in any direction facing a camera; and/or 2) a person can walk across multiple cameras which have different settings. An 'observed view' is an external factor which will not affect an individual walking manner. However, view change will significantly alter available visual gait information for matching. Recently, there are three possible tracks of solutions: 1) reconstructing 3D gait models; 2) seeking view-invariant gait features; 3) learning mapping/projection relationships of gaits across different views.

Second, in practice, a person can walk freely in any speed. A 'walking speed' is an internal factor which will directly affect an individual walking manner. When walking speed increases: 1) arms swing higher; 2) legs lift up higher; 3) stride length becomes longer; 4) *gait period* is shorter. Recently, there are two possible tracks of solutions: 1) seeking speed-invariant gait features; 2) learning mapping/projection relationships of gaits across different walking speeds.

This thesis has proposed five main methods for gait recognitions under various walking variations focusing on view changes and speed changes. These methods have been developed from various perspectives, which contain different benefits and limitations. The detailed summaries of the proposed methods are given in each conclusion section of each chapter. This section is to summarize them briefly in terms of key concepts, adopted techniques, advantages, and disadvantages.

First, the method in Chapter 3 is proposed for gait recognition under changes of various walking conditions such as view, walking speed, and carrying condition. **Key concept.** The binary pattern is generated as a robust gait feature by using binary derivatives to describe surrounding of a pixel in *GEI*. This process makes *GEI* more robust against uncertainties of gait information due to changes of walking conditions. **Adopted technique.** *Local Binary Pattern (LBP)* is applied to perform the binary derivatives. **Advantage.** It is robust to changes of various walking conditions. **Disadvantage.** Its performance drops when changes of walking conditions become significant. This challenge can be addressed by using the proposed methods in Chapters 4, 5, 6, and 7 which directly address the problems of view change and speed change.

Second, the methods in Chapter 4 are proposed for gait recognition under view change. **Key concept.** *VTM* is used to transform gait from one view into another view. The methods are proposed based on two types of *VTM* (i.e. *dVTM* and *rVTM*). *dVTM* is to decompose gait information into subject-independent and view-independent components. Such subject-independent component is dependent to view factor, which is used to project gait across different views. On the other hand, *rVTM* is to bridge correlated walking motions across different views based on regression models. **Adopted technique.** *Singular Value Decomposition (SVD)* is applied for the gait matrix decomposition in dVTM construction. *Multi-Layer Perceptron (MLP)*, *Support Vector Regression (SVR)*, and *Sparse Regression (SR)* are used as regression tools in rVTM construction. **Advantage.** 1) It is applicable for both *cross-view and multi-view gait recognitions*. 2) Its performance is very high and reliable, particularly when view change is not too significant (e.g. $\leq 36°$). **Disadvantage.** 1) It is not applicable for untrained/unseen views. The practical solution is given in Section 4.10. Moreover, this limitation can also be avoided by using the proposed method in Chapter 7 which does not rely on supervised learning and thus is applicable for unseen view. 2) Its performance is not fully satisfied when view change is very large. The practical solutions are given in Sections 4.8.2 and 4.8.3. This limitation can also be avoided by using the proposed method in Chapter 5 which relies on segment (i.e. group of pixels) mapping and thus is more robust to small/none correlated features of gaits from largely different views.

Third, the method in Chapter 5 is proposed for gait recognition under view change. **Key concept.** Gaits from different views are co-clustered into correlated segments. Each pair of such correlated segments across different views is projected into a common subspace through the processes of linear correlation optimization and *linear approximation*. **Adopted technique.** *Bipartite graph multipartitioning* is applied for co-clustering of gaits from different views. *Canonical Correlation Analysis (CCA)* is applied for optimization of correlation between gaits from different views. **Advantage.** 1) It is applicable for both *cross-view and multi-view gait recognitions*. 2) It is also applicable for *multi-view to multi-view gait recognition*. 3) It outperforms other existing methods in the literature, especially for the case of large view change. **Disadvantage.** As similar to the proposed method in Chapter 4, the proposed method in Chapter 5 also relies on supervised learning. Thus, it is not applicable for untrained/unseen views. As mentioned above, this limitation can be addressed by using the practical solution given in Section 4.10 or can be avoided by using the proposed method in Chapter 7.

Fourth, the method in Chapter 6 is proposed for gait recognition under walking speed change. **Key concept.** Each gait shape is described using *HSC* which takes higher-order derivatives along the shape boundary. In this way, *HSC* well represents local-static gait information which is robust to walking speed change. Later, *HSC* is used further in the statistical shape analysis to generate a speed-invariant gait feature. The shape boundary is also decomposed into several segments where each segment covers gait parts that are similarly affected by speed change. This makes it more accurate in the process of statistical shape analysis for gait feature extraction and similarity measurement. **Adopted technique.** *Procrustes Shape Analysis (PSA)* is applied for gait feature extraction and similarity measurement. **Advantage.** 1) It is applicable for any walking speed. 2) It achieves very high performance and thus outperforms other existing methods in the literature, significantly for the case of large speed change.

Fifth, the method in Chapter 7 is proposed for gait recognition under view change. **Key concept.** A domain transformation representing a scene of gait under a certain view is obtained through low-rank textures analysis on its corresponding *gait texture image*. The domain transformation can then be used to normalize gait silhouettes in the scene onto the common canonical view. The view-normalized gait silhouettes are used further to generate a novel view-invariant gait feature based on the statistical shape analysis. **Adopted technique.** *Transform Invariant Low-rank Textures (TILT)* is applied for the view-normalization process. *Procrustes Shape Analysis (PSA)* is applied for gait feature extraction and similarity measurement. **Advantage.** 1) It is applicable for any trained/untrained view since it does not rely on supervised learning. 2) Its performance is high and reliable for the difficult case of large view change, especially on approximately side views. **Disadvantage.** It is not efficient for approximately frontal views which are too different from the canonical view (i.e. approximately side view). This is a common limitation for all methods in the second category of gait recognition under view change (see Section 1.2.2.2).

8.2 Future Works

The future works to improve each proposed method are discussed in the conclusion of each chapter. This section gives discussions of how to, logically and maturely, combine the proposed methods for more practical situations. Moreover, the discussions also include extensions of our study on gait to closely related research areas.

8.2.1 Combination of the proposed methods for more practical situations

8.2.1.1 Combination of the proposed methods of gait recognitions under view change and speed change

It can be seen that both proposed methods in Chapters 6 and 7 are based on *PSA*, but are with different additional processes for specific solving different problems of speed change and view change respectively. These two methods have a potential to be combined for developing a new view- and speed-invariant gait feature. This will make it more practical in the real-world when a person walks with different directions and speeds.

The proposed methods in Chapters 6 and 7 can be properly merged by combining the frameworks in Figures 6.7 and 7.2. As in Figure 7.2, the proposed method based on low-rank textures analysis is used to obtain the view-normalized gait silhouettes which will be used as an input of the proposed method in Figure 6.7. In Chapter 6, the proposed method in Figure 6.7 is used to generate the speed-invariant gait feature based on *PSA*. However, by using the view-normalized gait silhouettes as the input, it can be used to generate a new gait feature which will be robust to both speed and view changes.

Moreover, *HSC* (in Chapter 6) will be used as a shape descriptor instead of *PSC* (in Chapter 7) because *PSC* is equivalent to HSC_1. The first-order derivatives of the shape boundary in PSC/HSC_1 is usually sufficient to tolerate view change. However, the higher-order derivatives are required to address the problem of speed change. Thus, in this future work, the order of derivative in *HSC* will be traded off between a degree of view change and a degree of speed change. As mentioned, the higher order of derivative will contain less gait information although it will be more robust to walking variations (see Figures 6.8, 6.9, 6.10). In addition, *DCM* will be constructed in the same way as it does for speed change in Chapter 6, but will take into account both view and speed changes.

The current difficulty is that there is not any database available to verify this work. The database must contain gaits of individuals under both variations of view and speed. Such database will be collected in the future.

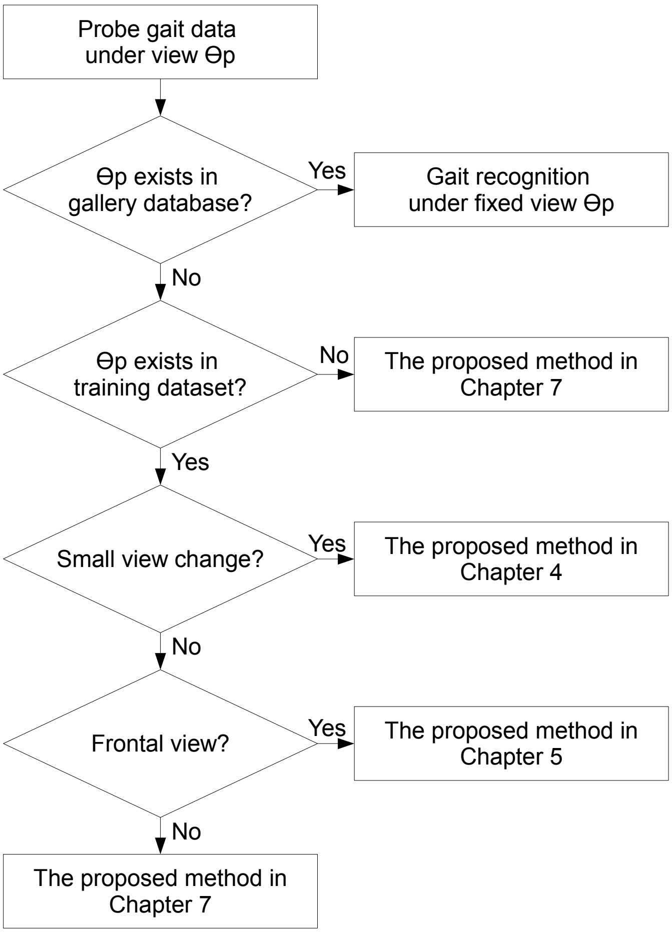


Figure 8.1: Combination of the proposed methods of gait recognition under view change.

8.2.1.2 Combination of the proposed methods of gait recognition under view change.

As mentioned that the proposed methods have been developed from various perspectives to address diverse aspects of gait recognition under view change, they can be logically combined to build up a more efficient solution. This can be achieved by considering advantages and disadvantages of each proposed method, as illustrated in Figure 8.1.

As can be seen in Figure 8.1, when view θ_p of probe gait data exists in gallery database, gait recognition can be directly carried out under the fixed view θ_p.

Otherwise, it means that views (θ_p, θ_g) of probe gait data and gallery gait data are different. Thus, view-normalization process will be required.

When θ_p and θ_g do not exist in the training dataset which is used to construct view-normalization models, the proposed method in Chapter 7 will be adopted because it can generate view-invariant gait feature without requiring any supervised learned model.

On the other hand, when θ_p and θ_g exist in the training dataset and the view difference $|\theta_p$ - $\theta_g|$ is small, the proposed method in Chapter 4 will be adopted since it can achieve very accurate and reliable performance. However, when the view change is relatively large, the proposed method in Chapter 7 will be adopted for the case of approximately side view. Otherwise, the proposed method in Chapter 5 will be adopted. This is because the proposed method in Chapter 7 can achieve very high performance for the case of large view change under approximately side view, while the proposed method in Chapter 5 can achieve fine performance for the case of large view change under any trained view.

In the future work, this proposed system will be verified in the real-world surveillance application.

8.2.2 Combination of gait with other biometrics

Rather than gait, human can also be identified by using other biometrics such as face, iris, finger-print, palm-print, and voice. Among these biometrics, gait and face can be simultaneously obtained by most surveillance systems. Thus, they can be combined to achieve a more reliable human identification in surveillance applications. Many methods on fusion of gait and face for human identification have been published recently based on three strategies.

First, the fusion is operated in feature level [156][157][158]. The individual face and gait features are normalized to have their values lie within similar ranges. Then, they are concatenated to form feature vectors based on characteristics of both face and gait, by using statistical tools such as Multiple Discriminant Analysis (MDA), *Principal Component Analysis (PCA)*, and *Linear Discriminant Analysis (LDA)*.

Second, the fusion is operated in match score level by using static fusion rules [7][39][159][160][161][162]. Before combining match scores of face and gait classifiers, it is necessary to normalize them onto same range of values by using analytical tool such as Gaussian model. Then, they can be fused based on different static fusion rules including sum, product, min, max, hierarchical fusion, and Bayesian decision.

Third, the fusion is operated in match score level by using adaptive fusion rules which can be dynamically adjusted to suit real-time external conditions [110][163]. As a typical example, adaptive fusion of gait and face can be driven by using a subject-to-camera distance and an observed view. That is, face is more reliable when a person is closer to a camera, while it is less efficient than gait when a person is relatively far from a camera. Moreover, face is most informative when it is observed under frontal view, while the best view to recognize gait is side view. Also, face is totally lost under rear view.

In the future work, based on our study on gait recognition under various walking variations, it can be said that quality of gait for recognizing human drops when there is a difference between walking conditions of probe and gallery gaits. Thus, in practical situations, an ability of recognizing gait/face across conditions will be also taken into a consideration for adaptive fusion of gait and face. For example, cross-view face/gait recognition of views between $90°$ and $108°$ is more reliable than views between $90°$ and $126°$. Also, gait may be more reliable than face under same degree of view change.

Moreover, there are other factors that also have impacts on weighting of face and gait in the fusion, including an image resolution and a shadow. An image resolution may affect face more than gait. In contrast, a shadow may affect gait more than face.

In the future work, these factors will be carefully used to build up a multi-factors model for fusing gait and face in match score level for human identification.

8.2.3 Gait analysis for other research fields

Currently, most studies (including this thesis) on gait focus on gait recognition for human identification. However, gait can also be used in other research fields such as aged care [134][164][165][166], health science [167][168][169][170][171], forensics and security [172][173][174], sport science [175][176][177], age estimation [178][179][180], and gender classification [181][182][183].

8.2.3.1 Aged care

Home accidents such as accidental stumbles and falls are major sources of morbidity and mortality among elders. Recently, gait monitoring system is used to help caregivers to detect and/or prevent these incidents.

In the future work, our study in this thesis can be extended and applied on home monitoring for aged care. Specifically, a method will be proposed to detect near falls and/or walking patterns that may cause injury. Our current research based on speed variations will be further studied and used for this purpose. Actions leading to injury can be caused by improper manners of walking speed, arms' and legs' lifting, stride length, joints' angle, etc. These gait qualities will be measured. Then, risky actions will be warned to gain more attention from a caregiver.

8.2.3.2 Health science

An unsteady gait is an abnormality in walking that can be caused by diseases, injuries of legs and feet, and/or damages to nervous system that controls movements necessary for walking. For example, it has been found that walking disorders (e.g. decreases of walking speed, cadence, and stride length) may be early signs of cognitive health decline such as Alzheimer's disease and degenerative neurological disorder such as Parkinson's disease.

Based on our reviews, it can be seen that gait is enormously related to health science. In the future work, we will propose an automatic and intelligent system for clinical usages based on gait analysis. Our works in this thesis based on walking variations and their impacts on walking patterns will be further studied to detect abnormal gaits, and then used to automatically indicate diseases. Moreover, early detection of abnormalities will also help to prevent physical injuries of proximal joints such as knee and hip.

In addition, gait analysis will be newly applied to improve a quality of physical therapy. It can be used for monitoring patients to perform the therapy properly which will help them to recover faster. This idea will also include seeking of specific areas which may require more attention.

8.2.3.3 Forensics and security

In the research area of forensics and security, gait is majorly used for human identification. However, it can also be used to detect suspicious actions such as fighting, robbery, and terrorist. Abnormal gait can be used to generate early warning of alarming actions in public areas such as airports and subway stations. For example, gait analysis can detect whether a person hides an heavy object (which can be weapon or bomb) under his jacket. In this case, a police officer may ask him to undergo security check.

In the future work, we will propose an automatic security system in public area by using gait analysis under multiple cameras. This system will be much more reliable than existing single camera-based systems. Our works of multi-view gait analysis in this thesis will be adopted and adapted to build up such system.

8.2.3.4 Sport science

In sports, gait analysis can be used to improve athlete's performance, prevent injury, and optimize enjoyment. Gait analysis examines aspects and qualities of muscle length, joint mobility, joint stability, motor control and coordination, appropriate muscle action, resiliency and resistance to fatigue, and functional strength. This provides a progressive and objective view of where weak links, asymmetries, and other compensations exist. These compensations in movement pattern can ultimately lead to poor biomechanics, lower efficiency and power, and even injury.

In the future work, we will propose a system based on gait analysis to help athletes become stronger, faster, and more injury-resistant. The system will also look at various methods used to assess posture and stability in athletes. From this thesis, our analyses of walking speed change and its impacts on gaits will be used in sport science. For example, a right speed of movement can result in a proper action which will lead to an optimized athlete's performance.

8.2.3.5 Age estimation

It has been proved that gait analysis can be used to approximately estimate human age. The existing methods rely on statistical analyses without considering actual relations between age and gait. However, there are scientific evidences [184][185] showing that age and gait are strongly related. For example, it has been demonstrated that a person walks slower with shorter stride length when he gets older.

Therefore, our works based on walking speed change and its impacts on gaits (e.g. stride length, arms' and legs' swing) will be applied to improve estimation of human age.

8.2.3.6 Gender classification

It has been proved that human gaits vary between males and females although they are in a same age group. The analysis in [186] has shown that females walk with lesser step width and more pelvic movement. Moreover, females walk with hip sway, while males walk with swagger in shoulder. The main kinematics-differences between male and female walking patterns are dependent to hip joint.

Based on our study in this thesis and scientific evidence in [186], it can be said that some walking manners including stride length, arms' swing, and legs' lifting are dependent to both walking speed and gender. Thus, our study will be further extended to make gender classification based on gait analysis more robust to various walking speeds.

References

[1] L. Wang, G. Zhao, N. Rajpoot, and M.S. Nixon. Special issue on new advances in video-based gait analysis and applications: challenges and solutions. *IEEE Tran. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 40(4):982–985, August 2010. 1

[2] S. Sarkar, P. J. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. W. Bowyer. The humanid gait challenge problem: data sets, performance, and analysis. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 27(2):162–177, February 2005. 1, 15, 45, 96, 105, 119, 120, 156, 181

[3] S. Yu, D. Tan, and T. Tan. A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition. pages 441–444, Hong Kong, August 2006. IEEE Int. Conf. on Pattern Recognition. 2, 14, 90, 91, 113, 114, 117

[4] C. BenAbdelkader. Gait as a biometric for person identification in video, 2002. Ph.D. thesis in Maryland university. 3

[5] S. Yu, D. Tan, and T. Tan. Modelling the effect of veiw angle variation on apperance-based gait recognition. pages 807–816, India, January 2006. Asian Conf. on Computer Vision. 3

[6] X. Geng, K. Smith-Miles, L. Wang, M. Li, and Q. Wu. Context-aware fusion: a case study on fusion of gait and face for human identification in video. *Pattern Recognition*, 43(10):3660–3673, October 2010. 4, 158, 159

[7] X. Zhou and B. Bhanu. Integrating face and gait for human recognition at a distance in video. *IEEE Tran. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 37(5):1119–1137, October 2007. 4, 158, 159, 189

[8] S. R. Das, R. C. Wilson, M. T. Lazarewicz, and L. H. Finkel. Gait recognition by two-stage principal component analysis. pages 579–584, United States, April 2006. Int. Conf. on Automatic Face and Gesture Recognition. 4

[9] A. Tsuji, Y. Makihara, and Y. Yagi. Silhouette transformation based on walking speed for gait identification. pages 717–722, United States, June 2010. IEEE Conf. on Computer Vision and Pattern Recognition. 5, 13, 16, 138, 151, 152, 154

[10] R. C. Browning, E. A. Baker, J. A. Herron, and R. Kram. Effects of obesity and sex on the energetic cost and preferred speed of walking. *Journal of Applied Physiology*, 100(2):390–398, October 2006. 5

[11] R. V. Levine and A. Norenzayan. The pace of life in 31 countries. *Journal of Cross-Cultural Psychology*, 30(2):178–205, March 1999. 5

[12] B. J. Mohler, W. B. Thompson, S. H. Creem-Regehr, H. L. Jr. Pick, and W. H. Jr. Warren. Visual flow influences gait transition speed and preferred walking speed. *Experimental Brain Research*, 181(2):221–228, March 2007. 5

[13] A. E. Minetti. *Chapter 5: The three modes of terrestrial locomotion, in Biomechanics and biology of movement*, pages 67–78. Human Kinetics, United States, 2000. 5

[14] D. Rose. *A comprehensive balance and mobility training program*. Human Kinetics, United States, 2 edition, 2010. 5, 120, 135

[15] C. B. Abdelkader, R. Culter, and L. Davis. Stride and cadence as a biometric in automatic person identification and verification. pages 372–377, United States, May 2002. Int. Conf. on Automatic Face and Gesture Recognition. 6

[16] A. Bobick and A. Johnson. Gait recognition using static, activity-specific parameters. pages 423–430, United States, December 2001. IEEE Int. Conf. on Computer Vision and Pattern Recognition. 6, 7

[17] I. Bouchrika and M. S. Nixon. Model-based feature extraction for gait analysis and recognition. pages 150–160, France, March 2007. Int. Conf. on Computer Vision/Computer Graphics Collaboration Techniques and Applications. 6, 7

[18] D. Cunado, M. S. Nixon, and J. N. Carter. Using gait as a biometric, via phase-weighted magnitude spectra. pages 95–102, Switzerland, March 1997. Int. Conf. on Audio- and Video-Based Biometric Person Authentication. 6

[19] N. Huazhong, T. Tieniu, W. Liang, and H.Weiming. Kinematics-based tracking of human walking in monocular video sequences. *Journal of Image and Vision Computing*, 22(5):429–441, May 2004. 6, 161

[20] A. Johnson and A. Bobick. A multi-view method for gait recognition using static body parameters. pages 301–311, Sweden, June 2001. Int. Conf. on Audio- and Video-Based Biometric Person Authentication. 6, 161

[21] C. S. Lee and A. Elgammal. Gait tracking and recognition using person-dependent dynamic shape model. pages 553–559, United Kingdom, April 2006. Int. Conf. on Automatic Face and Gesture Recognition. 6, 7

[22] L. Wang, T. Tan, H. Ning, and W. Hu. Fusion of static and dynamic body biometrics for gait recognition. *IEEE Tran. on Circuits and Systems for Video Technology*, 14(2):149–158, February 2004. 6

[23] C. Y. Yam and M. S. Nixon. *Model-based gait recognition*. Encyclopedia of Biometrics. Springer, 2009. 6, 7

[24] C. Y. Yam, M. S. Nixon, and J. N. Carter. Gait recognition by walking and running: A model-based approach. pages 1–6, Australia, 2002. Asian Conf. on Computer Vision. 6

[25] C. Chen, J. Liang, H. Zhao, H. Hu, and J. Tian. Frame difference energy image for gait recognition with incomplete silhouettes. *Pattern Recognition Letters*, 30(11):977–984, August 2009. 6

[26] S. D. Choudhury and T. Tjahjadi. Silhouette-based gait recognition using procrustes shape analysis and elliptic fourier descriptors. *Pattern Recognition*, 45(9):3414–3426, September 2012. 6

[27] J. Han and B. Bhanu. Individual recognition using gait energy image. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 28(2):316–322, February 2006. 6, 7, 23, 41, 44, 58, 61, 96, 119, 156, 175, 179, 181

[28] Z. Liu and S. Sarkar. Improved gait recognition by gait dynamics normalization. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 28(6): 863–876, 2006. 6, 7, 12, 96, 119, 155, 156, 181

[29] A. Roy, S. Sural, and J. Mukherjee. Gait recognition using pose kinematics and pose energy image. *Signal Processing*, 92(3), March 2012. 6, 7, 119, 181

[30] D. Tan, K. Huang, S. Yu, and T. Tan. Uniprojective features for gait recognition. pages 673–682, Korea, August 2007. Int. Conf. on Biometrics. 6, 7, 12, 52, 55, 58, 150

[31] I. Venkat and P. D. Wilde. Robust gait recognition by learning and exploiting sub-gait characteristics. *Int. Journal of Computer Vision*, 91(1):7–23, 2011. 6

[32] C. Wang, J. Zhang, L. Wang, J. Pu, and X. Yuan. Human identification using temporal information preserving gait template. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, December 2011. 6, 7, 160

[33] Z. Xue, D. Ming, W. Song, B. Wan, and S. Jin. Infrared gait recognition based on wavelet transform and support vector machine. *Pattern Recognition*, 43(8):2904–2910, August 2010. 6

[34] E. Zhang, Y. Zhao, and W. Xiong. Active energy image plus 2dlpp for gait recognition. *Signal Processing*, 90(7):2295–2302, July 2010. 6, 150

[35] C. BenAbdelkader, R. Culter, H. Nanda, and L. Davis. Eigengait: motion-based recognition of people using image self-similarity. pages 284–294, Sweden, June 2001. Int. Conf. on Audio- and Video-Based Biometric Person Authentication. 7, 53

[36] C. BenAbdelkader, R. Culter, and L. Davis. Motion-based recognition of people in eigengait space. pages 267–272, United States, May 2002. Int. Conf. on Automatic Face and Gesture Recognition. 7, 53

[37] Y. Chai, Q. Wang, R. Zhao, and C. Wu. A new automatic gait recognition method based on the perceptual curve. pages 1–5. IEEE Int. Conf. on Intelligent Information Communication Technologies for Better Human Life, November 2005. 7, 55

[38] R. Bodor, A. Drenner, D. Fehr, O. Masoud, and N. Papanikolopoulos. View-independent human motion classification using image-based reconstruction. *Journal of Image and Vision Computing*, 27(8):1194–1206, July 2009. 8

[39] G. Shakhnarovich, L. Lee, and T. Darrell. Integrated face and gait recognition from multiple views. pages 439–446, United States, December 2001. IEEE Conf. on Computer Vision and Pattern Recognition. 8, 189

[40] Z. Zhang and N. F. Troje. View-independent person identification from human gait. *Journal of Neurocomputing: Neural Networks in Signal Processing*, 69(1-3):250–256, December 2005. 8

[41] G. Zhao, G. Liu, H. Li, and M. Pietikainen. 3d gait recognition using multiple cameras. pages 529–534, United Kingdom, April 2006. IEEE Int. Conf. on Automation Face Gesture Recognition. 8, 94

[42] R. D. Seely, S. Samangooei, M. Lee, J. N. Carter, and M. S. Nixon. The university of southampton multi-biometric tunnel and introducing a novel 3d gait dataset. pages 1–6, United States, September 2008. IEEE Int. Conf. on Biometrics: Theory, Applications and Systems. 9

[43] W. Kusakunniran, H. Li, and J. Zhang. A direct method to self-calibrate a surveillance camera by observing a walking pedestrian. pages 250–255, Australia, December 2009. Digital Image Computing: Techniques and Applications. 9, 73

[44] M. Goffredo, I. Bouchrika, J. Carter, and M. Nixon. Self-calibrating view-invariant gait biometrics. *IEEE Tran. on Systems, Man and Cybernetics B*, 40(4):997–1008, October 2009. 9, 90, 91, 113, 114, 117, 157, 175, 179

[45] J. Han, B. Bhanu, and A. K. Roy-Chowdhury. A study on view-insensitive gait recognition. pages 297–300, Italy, September 2005. IEEE Int. Conf. on Image Processing. 9

[46] F. Jean, R. Bergevin, and A. B. Albu. Computing and evaluating view-normalized body part trajectories. *Journal of Image and Vision Computing*, 27(9):1272–1284, August 2009. 9

[47] A. Kale, K. R. Chowdhury, and R. Chellappa. Towards a view invariant gait recognition algorithm. pages 143–150, United States, July 2003. IEEE Conf. on Advanced Video and Signal Based Surveillance. 9

[48] K. Bashir, T. Xiang, and S. Gong. Cross-view gait recognition using correlation strength. United Kingdom, August 2010. British Machine Vision Conference. 10, 11, 41, 65, 67, 85, 86, 90, 91, 97, 99, 100, 101, 113, 114, 115, 117, 119, 176, 180

[49] W. Kusakunniran, Q. Wu, J. Zhang, and H. Li. Support vector regression for multi-view gait recognition based on local motion feature selection. pages 974–981, United States, June 2010. IEEE Int. Conf. on Computer Vision and Pattern Recognition. 10, 11, 13, 22, 89, 97, 99, 100, 101, 113, 114, 115, 117, 118, 136, 176, 180

[50] W. Kusakunniran, Q. Wu, H. Li, and J. Zhang. Multiple views gait recognition using view transformation model based on optimized gait energy image. pages 1058–1064, Japan, September-October 2009. IEEE Int. Conf. on Computer Vision (THEMIS workshop). 10, 11, 13, 22, 45, 97, 113, 114, 115, 117, 118, 176, 180

[51] N. Liu and Y. P. Tan. View invariant gait recognition. pages 1410–1413, United States, March 2010. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing. 10, 11, 83, 84, 94, 97

[52] Y. Makihara, R. Sagawa, Y. Mukaigawa, T. Echigo, and Y. Yagi. Gait recognition using a view transformation model in the frequency domain. pages 151–163, Austria, July 2006. European Conf. on Computer Vision. 10, 11, 13, 18, 41, 61, 89, 90, 97, 113, 117, 136, 176, 180

[53] W. Kusakunniran, Q. Wu, H. Li, and J. Zhang. Automatic gait recognition using weighted binary pattern on video. pages 49–54, Italy, September 2009. IEEE Conf. on Advanced Video and Signal Based Surveillance. 13, 22, 150

[54] R. Tanawongsuwan and A. Bobick. Modelling the effects of walking speed on appearance-based gait recognition. pages 783–790, United States, June 2004. IEEE Int. Conf. on Computer Vision and Pattern Recognition. 13, 138, 151, 152, 153, 154

[55] R. Tanawongsuwan and A. Bobick. Performance analysis of time-distance gait parameters under different speeds. pages 715–724, United Kingdom, June 2003. Int. Conf. on Audio- and Video-based Biometric Person Authentication. 13

[56] D. Tan, K. Huang, S. Yu, and T. Tan. Efficient night gait recognition based on template matching. pages 1000–1003, Hong Kong, August 2006. Int. Conf. on Pattern Recognition. 15, 58, 150

[57] R. Gross and J. Shi. The cmu motion of body (mobo) database technical report. *CMU-RI-TR-01-18*, Robotic Institute, Carnegie Mellon University, 2001. 16

[58] W. Kusakunniran, Q. Wu, J. Zhang, and H. Li. Cross-view and multi-view gait recognitions based on view transformation model using multi-layer perceptron. *Pattern Recognition Letters*, 33(7):882–889, May 2012. 22

[59] W. Kusakunniran, Q. Wu, J. Zhang, and H. Li. Gait recognition under various viewing angles based on correlated motion regression. *IEEE Tran. on Circuits and Systems for Video Technology*, 22(6):966–980, February 2012. 22

[60] W. Kusakunniran, Q. Wu, J. Zhang, and H. Li. Gait recognition across various walking speeds using higher-order shape configuration based on a differential composition model. *IEEE Tran. on Systems, Man, and Cybernetics, Part B: Cybernetics*, May 2012. 22

[61] W. Kusakunniran, Q. Wu, J. Zhang, and H. Li. Multi-view gait recognition based on motion regression using multilayer perceptron. pages 2186–2189, Turkey, August 2010. IEEE Int. Conf. on Pattern Recognition. 22

[62] W. Kusakunniran, Q. Wu, J. Zhang, and H. Li. Pairwise shape configuration-based psa for gait recognition under small viewing angle change. pages 17–22, Austria, August 2011. IEEE Conf. on Advanced Video and Signal Based Surveillance. 22, 159, 167, 175, 179

[63] W. Kusakunniran, Q. Wu, J. Zhang, and H. Li. Speed-invariant gait recognition based on procrustes shape analysis using higher-order shape configuration. pages 545–548, Belguim, September 2011. IEEE Int. Conf. on Image Processing. 22

[64] X. Zou and B. Bhanu. Human activity classification based on gait energy image and coevolutionary genetic programming. pages 556–559, Hong Kong, September 2006. Int. Conf. on Pattern Recognition. 23, 61

[65] L. Zhang, S. Z. Li, X. Yuan, and S. Xiang. Real-time object classification in video surveillance based on appearance learning. pages 1–8, United States, June 2007. IEEE Conf. on Computer Vision and Pattern Recognition. 25, 48

[66] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1):51–59, January 1996. 25, 48

[67] H. Abdi and L. J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010. 26

[68] S. Fidler and A. Leonardis. Robust lda classification by subsampling. pages 97–104, United States, June 2003. Int. Workshop on Statistical Analysis in Computer Vision. 26, 27, 81

[69] K. Fukunaga. *Introduction to statistical pattern recognition*. Elsevier, 2 edition, 1990. 27

[70] J. Ye. Least squares linear discriminant analysis. United States, June 2007. Int Conf. on Machine Learning. 27

[71] L. N. Trefethen and D. Bau III. *Numerical linear algebra*. SIAM: Society for Industrial and Applied Mathematics, 1 edition, 1997. 27

[72] V. Klema and A. Laub. The singular value decomposition: its computation and some applications. *IEEE Tran. on Automatic Control*, 25(2):164–176, April 1980. 27

[73] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. J. Wiley, New York, United States, 2001. 29, 30, 76

[74] N. A. Thacker. Tutorial: supervised neural networks in machine vision. the Image Science and Biomedical Engineering Division, Medical School, University of Manchester, December 1998. 29, 76

[75] R. Ebrahimpour, E. Kabir, and M. R. Yousefi. Face detection using mixture of mlp experts. *Neural Processing Letters*, 26(1):69–82, June 2007. 29

[76] F. Smach, M. Atri, J. Miteran, and M. Abid. Design of a neural networks classifier for face detection. *World Academy of Science, Engineering and Technology*, 11:124–127, November 2005. 29

[77] A. J. Smola and B. Scholkopf. A tutorial on support vector regression. *Journal of Statistics and Computing*, 14(3):199–222, August 2004. 30, 31

[78] N.A. Sakhanenko, G.F. Luger, H.E. Makaruk, J.B. Aubrey, and D.B. Holtkamp. Shock physics data reconstruction using support vector regression. *International Journal of Modern Physics C*, 17(9):1313–1325, June 2006. 30

[79] W. Zhu, Y. Chen, Y. Sun, B. Yin, and D. Jiang. Svr-based facial texture driving for realistic expression synthesis. pages 456–459, Hong Kong, December 2004. Int. Conf. on Image and Graphics. 30

[80] C. Fan, H. Sarrafzadeh, F. Dadgostar, and H. Gholamhosseini. Facial expression analysis by support vector regression. pages 29–34, New Zealand, 2005. University of Otago. 30

[81] A. Ben-Hur, C. Ong, S. Sonnenburg, B. Scholkopf, and G. Ratsch. Support vector machines and kernels for computational biology. *Public Library of Science Computational Biology: Journal of the International Society for Computational Biology*, 4(10):1–10, October 2008. 33

[82] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996. 33, 34

[83] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005. 33, 34, 79

[84] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. *Lease angle regression.* The annals of statistics 32, In press, 2004. 34

[85] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, June 2006. 34

[86] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. pages 269–274, United States, 2001. Int. Conf. on Knowledge Discovery and Data Mining. 35

[87] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000. 35

[88] D. R. Hardoon, S. Szedmak, and J. S. Taylor. Canonical correlation analysis: an overview with application to learning method. *Neural Computation*, 16(12):2639–2664, December 2004. 38

[89] W. Zheng, X. Zhou, C. Zou, and L. Zhao. Facial expression recognition using kernel canonical correlation analysis (kcca). *IEEE Tran. on Neural Networks*, 17(1):233–238, January 2006. 38

[90] J. Boyd. Video phase-locked loops in gait recognition. pages 696–700, Canada, July 2001. IEEE Int. Conf. on Computer Vision. 39, 122, 127, 169

[91] I. Dryden and K. Mardia. *Statistical Shape Analysis*. New York: John Wiley & Sons, 1999. 39, 41

[92] A. Ericsson. Automatic shape modeling with applications in medical imaging. *PhD thesis*, Lund University, Centre for Mathematical Sciences, Box 118, SE-22100, Lund, Sweden, 2006. 39, 122, 128

[93] Z. Zhang, A. Ganesh, X. Liang, and Y. Ma. Tilt: Transform invariant low-rank textures. *Int. Journal of Computer Vision*, pages 1–24, January 2012. 41, 43, 164, 165, 178

[94] E. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11:1–11:37, May 2011. 43, 165

[95] A. M. Elgammal, D. Harwood, and L. S. Davis. Non-parametric model for background subtraction. pages 751–767, Ireland, June 2000. European Conf. on Computer Vision. 45, 161

[96] J. Liang, C. Chen, H. Zhao, H. Hu, and J. Tian. *Chapter 9: Gait Feature Fusion using Factorial HMM, Behavioral Biometrics for Human Identification: Intelligent Applications*. IGI Global, United States, 2010. 45, 162

[97] L. Wang, T. Tan, H. Ning, and W. Hu. Silhouette analysis based gait recognition for human identification. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 25(12):1505–1518, December 2003. 45, 46

[98] A. Mori, Y. Makihara, and Y. Yagi. Gait recognition using period-based phase synchronization for low frame-rate videos. pages 2194–2197, Turkey, August 2010. Int. Conf. on Pattern Recognition. 45, 47

[99] T. H. W. Lam, K.H. Cheung, and J. N. K. Liu. Gait flow image: a silhouette-based gait representation for human identification. *Pattern Recognition*, 44(4):973–987, April 2011. 45, 47

[100] N. V. Boulgouris, K. N. Plataniotis, and D. Hatzinakos. Gait recognition using dynamic time warping. pages 263–266, Canada, October 2004. IEEE Workshop on Multimedia Signal Processing. 45, 47

[101] X. Ben, W. Meng, and R. Yan. Dual-ellipse fitting approach for robust gait periodicity detection. *Neurocomputing*, 79:173–178, March 2012. 45, 47

[102] G. R. Arce. *Nonlinear signal processing: a statistical approach*. Wiley:New Jersey, United States, 2005. 46

[103] R. Collins, R. Gross, and J. Shi. Silhouette-based human identification from body shape and gait. pages 366–371, United States, May 2002. Int. Conf. on Automatic face and Gesture Recognition. 53, 155

[104] L. Lee and W. Grimson. Gait analysis for recognition and classification. pages 148–155, United States, May 2002. Int. Conf. on Automatic face and Gesture Recognition. 53, 155

[105] P. J. Phillips, S. Sarkar, I. Robledo, P. Grother, and K. Bowyer. Baseline results for the challenge problem of human id using gait analysis. pages 137–142, United States, May 2002. Int. Conf. on Automatic face and Gesture Recognition. 53

[106] L. Wang, T. Tan, W. Hu, and H. Ning. Automatic gait recognition based on statistical shape analysis. *IEEE Tran. on Image Processing*, 12(9):1120–1131, September 2003. 53, 121, 122, 127, 147, 169, 175, 179

[107] H. Su and F. Huang. Gait recognition using principal curves and neural networks. pages 238–243. Int. Conf. on Advnaces in Neural Networks, 2006. 53

[108] J. Lu, E. Zhang, and C. Jing. Gait recognition using wavelet descriptors and independent component analysis. pages 232–237. Int. Conf. on Advnaces in Neural Networks, 2006. 53

[109] Q. Wu, L. Wang, X. Geng, M. Li, and X. He. Dynamic biometrics fusion at feature level for video based human recognition. pages 152–157, New Zealand, December 2007. Int. Conf. on Image and Vision Computing New Zealand. 53

[110] X. Geng, L. Wang, M. Li, Q. Wu, and K. Smith-Miles. Distance-driven fusion of gait and face for human identification in video. pages 19–24, New Zealand, December 2007. Int. Conf. on Image and Vision Computing New Zealand. 53, 189

[111] C. P. Shi, H. G. Li, X. Lian, and X. G. Li. Multi-resolution local moment feature for gait recognition. pages 3709–3714. Int. Conf. on Machine Learning and Cybernetics, August 2006. 55

[112] F. Dadashi, B.N. Araabi, and H. Soltanian-Zadeh. Gait recognition using wavelet packet silhouette representation and transductive support vector machines. pages 1–5. Int. Cong. on Image and Signal Processing, October 2009. 58, 150

[113] D. Tan, K. Huang, S. Yu, and T. Tan. Orthogonal diagonal projections for gait recognition. pages 337–340. IEEE Int. Conf. on Image Processing, November 2007. 58, 150

[114] D. Tan, S. Yu, K. Huang, and T. Tan. Walker recognition without gait cycle estimation. pages 222–231. Int. Conf. on Biometrics, 2007. 58, 150, 155

[115] B. DeCann and A. Ross. Gait curves for human recognition, backpack detection and silhouette correction in a nighttime environment. United States, April 2010. SPIE Conf. on Biometric Technology for Human Identification. 58, 150

[116] J. Yang and J. Y. Yang. Why can lda be performed in pca transformed space? *Pattern Recognition*, 36(2):563–566, February 2003. 66

[117] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 19(7):711–720, July 1997. 66

[118] C. Liu and H. Wechsler. Robust coding schemes for indexing and retrieval from large face database. *IEEE Tran. on Image Processing*, 9(1):132–137, January 2000. 66

[119] B. S. Venkatesh, S. Palanivel, and B. Yegnanarayana. Face detection and recognition in an image sequence using eigenedginess. India, December 2002. Conf. on Computer Vision, Graphics Image Processing. 66, 81

[120] J. H. Yang and M. S. Yang. A control chart pattern recognition system using a statistical correlation coefficient method. *Journal of Computers and Industrial Engineering*, 48(2):205–221, March 2005. 67, 75

[121] A. Sharifinejad. Utilizing autoregressive truncated singular value decomposition algorithm for obtaining more efficiently compressed images. New Zealand, November 2003. Int. Conf. on Image and Vision Computing New Zealand. 70

[122] F. Lv, T. Zhao, and R. Nevatia. Camera calibration from video of a walking human. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(9): 1513–1518, September 2006. 73

[123] S. Shan, B. Cao, Y. Su, L. Qing, X. Chen, and W. Gao. Unified principal component analysis with generalized covariance matrix for face recognition.

pages 1–7, United States, June 2008. IEEE Int. Conf. on Computer Vision and Pattern Recognition. 73

[124] V. Cherkassky and F. Mulier. *Learning from data: Concepts theory and methods*. Wiley:New York, United States, 1998. 77, 78

[125] S. Ali and K. A. Smith. Automatic parameter selection for polynomial kernel. pages 243–249, United States, October 2003. IEEE Conf. on Information Reuse and Integration. 78

[126] W. Wang, Z. Xu, W. Lu, and X. Zhang. Determination of the spread parameter in the guassian kernel for classification and regression. *Neurocomputing*, 55(3):643–663, October 2003. 78

[127] T. Joachims. Svm-light support vector machine, 2008. version 6.02. 78

[128] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An efficient k-means clustering algorithm: analysis and implementation. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 24(7):881–892, July 2002. 84, 93

[129] D. Ioannidis, D. Tzovaras, I. G. Damousis, S. Argyropoulos, and K. Moustakas. Gait recognition using compact feature extraction transforms and depth information. *IEEE Tran. on Information Forensics and Security*, 2 (3):623–630, 2007. 96, 119, 156, 181

[130] T. H. W. Lam, R. S. T. Lee, and D. Zhang. Human gait recognition by the fusion of motion and static spatio-temporal templates. *Pattern Recognition*, 40(9):2563–2573, 2007. 96, 119, 156, 181

[131] X. Yang, J. Dai, Y. Zhou, and J. Yang. Gabor-based discriminative common vectors for gait recognition. pages 191–195, China, May 2008. Int. Congress on Image and Signal Processing. 96

[132] Q. Zhang and S. Xu. Gait-based recognition of human using an embedded hidden markov models. pages 1–4, China, December 2009. Int. Conf. on Information Engineering and Computer Science. 96, 119, 156, 181

[133] J. M. Lattin, J. D. Carroll, and P. E. Green. *Analyzing multivariate data*. Thomson Brooks/Cole, 2003. 109

[134] C. Rougier, J. meunier, A. St-Arnaud, and J. Rousseau. Procrustes shape analysis for fall detection. France, October 2008. Int. Workshop on Visual Surveillance. 121, 122, 169, 190

[135] L. Igual and F. De la Torre. Continuous procrustes analysis to learn 2d shape models from 3d objects. United States, June 2010. Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment. 121, 122

[136] W. Wu, Q. Guo, D. L. Massart, C. Boucon, and S. de Jong. Structure preserving feature selection in parafac using a genetic algorithm and procrustes analysis. *Journal of Chemometrics and Intelligent Laboratory Systems*, 65 (1):83–95, January 2003. 122

[137] D. Gonzalez-Arjona, G. Lopez-Perez, and A. G. Gonzalez. Holmes, a program for performing procrustes transformations. *Journal of Chemometrics and Intelligent Laboratory Systems*, 57(2):133–137, July 2001. 122

[138] W. H. Cho, S. W. Kim, M. E. Lee, S. H. Kim, S. Y. Park, and C. B. Jeong. Multimodality image registration using spatial procrustes analysis and modified conditional entropy. *Journal of Signal Processing Systems*, 54 (1):101–114, January 2009. 122

[139] R. Larsen. L1 generalized procrustes 2d shape alignment. *Journal of Mathematical Imaging and Vision*, 31(2):189–194, 2008. 122

[140] C. de Boor. Divided differences. *Surveys in Approximation Theory*, 1:46–69, 2005. 122, 132, 140

[141] T. Haig, Y. Attikiouzel, and M. Alder. Border following: new definition gives improved borders. pages 206–211. IEE on Vision Image and Signal Processing, 1992. 123

[142] W. Kusakunniran, Q. Wu, J. Zhang, and H. Li. Robust gait recognition based on procrustes shape analysis of pairwise configuration. *Technical Report 1021*, School of Computer Science and Engineering, University of New South Wales, Australia, 2010. 124

[143] T. Gopal and V. Prasad. A novel approach to shape based image retrieval integrating adapted fourier descriptors and freeman code. *Int. Journal of Computer Science and Network Security*, 8(6):293–302, June 2008. 124

[144] Y. Zhang, N. Yang, W. Li, X. Wu, and Q. Ruan. Gait recognition using procrustes shape analysis and shape context. pages 256–265, China, September 2009. Asian Conf. on Computer Vision. 127, 169

[145] S. Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of Research and Development*, 4(1):66–82, 1960. 141, 144

[146] W. Skarbek. Discriminant analysis diagram for pattern recognition. pages 1–6. IEEE Int. Workshop on Imaging Systems and Techniques, 2007. 144

[147] S. K. Alessandro, A. Magnani, and S. P. Boyd. Robust fisher discriminant analysis. pages 659–666. Advances in Neural Information Processing Systems, 2006. 144

[148] D. Tan, K. Huang, S. Yu, and T. Tan. Recognizing night walkers based on one pseudoshape representation of gait. pages 1–8. IEEE Conf. on Computer Vision and Pattern Recognition, 2007. 150

[149] A. Kale, N. Cuntoor, B. Yegnanarayana, A. N. Rajagopalan, and R. Chellappa. Gait analysis for human identification. pages 706–714. Int. Conf. on Audio- and Video-Based Biometric Person Authentication, 2003. 155, 156, 181

[150] C. Chen, J. Liang, and X. Zhu. Gait recognition based on improved dynamic bayesian networks. *Pattern Recognition*, 44:988–995, 2011. 155

[151] A. Veeraraghavan, A. R. Chowdhury, and R. Chellappa. Role of shape and kinematics in human movement analysis. IEEE Conf. on Computer Vision and Pattern Recognition, 2004. 155

[152] C. BenAbdelkader, R. G. Cutler, and S. Davis. Gait recognition using image self-similarity. *EURASIP Journal on Applied Signal Processing*, 4: 572–585, 2004. 155

[153] S. Lee, Y. Liu, and R. Collins. Shape variation-based frieze pattern for robust gait recognition. pages 1–8. IEEE Conf. on Computer Vision and Pattern Recognition, 2007. 155

[154] A. Kale, A. Sundaresan, A. Rajagopalan, N. Cuntoor, A. RoyChowdhury, and V. Krueger. Identification of humans using gait. *IEEE Tran. on Image Processing*, 13(9):1163–1173, 2004. 155

[155] A. Veeraraghavan, A. Roy-Chowdhury, and R. Chellappa. Matching shape sequences in video with applications in human movement analysis. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 27(12):1896–1909, 2005. 155

[156] X. Zhou and B. Bhanu. Feature fusion of face and gait for human recognition at a distance in video. pages 529–532, Hong Kong, August 2006. IEEE Int. Conf. on Pattern Recognition. 188

[157] X. Zhou and B. Bhanu. Feature fusion of side face and gait for video-based human identification. *Pattern Recognition*, 41(3):778–795, March 2008. 188

[158] E. Hossain and G. Chetty. Person identity verification based on multimodal face-gait fusion. *Int. Journal of Computer Science and Network Security*, 11(6):77–86, June 2011. 188

[159] B. Bhanu and J. Han. *Match score level fusion of face and gait at a distance, in Human recognition at a distance in video*, pages 185–207. Springer London, 2011. 189

[160] A. Kale, A. Roy-chowdhury, and R. Chellappa. Fusion of gait and face for human identification. pages 901–904, Canada, May 2004. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing. 189

[161] G. Shakhnarovich and T. Darrell. On probabilistic combination of face and gait cues for identification. pages 169–174, United States, May 2002. IEEE Int. Conf. on Automatic Face and Gesture Recognition. 189

[162] Z. Liu and S. Sarkar. Outdoor recognition at a distance by fusing gait and face. *Image and Vision Computing*, 25(6):817–832, June 2007. 189

[163] X. Geng, L. Wang, M. Li, Q. Wu, and K. Smith-Miles. Adaptive fusion of gait and face for human identification in video. pages 1–6, United States, Jauary 2008. IEEE Workshop on Applications of Computer Vision. 189

[164] Y. Chen and Y. Lin. Indoor rfid gait monitoring system for fall detection. pages 207–212, Taiwan, November 2010. Int. Symposium on Aware Computing. 190

[165] A. Weiss, I. Shimkin, N. Giladi, and J. M. Hausdorff. Automated detection of near falls: algorithm development and preliminary results. *BMC Research Notes*, 3(62), March 2010. 190

[166] V. Bianchi, F. Grossi, G. Matrella, I. De Munari, and P. Ciampolini. Fall detection and gait analysis in a smart-home environment. *Int. Journal on the Fundamental Aspects of Technology to Serve the Ageing Society.*, 7(2), 2008. 190

[167] J. M. Wood, P. F. Lacherez, A. A. Black, M. H. Cole, M. Y. Boon, and G. K. Kerr. Postural stability and gait among older adults with age-related maculopathy. *Investigative Ophthalmology and Visual Science*, 50(1):482–487, April 2009. 190

[168] J. Verghese, R. B. Lipton, C. B. Hall, G. Kuslansky, M. J. Katz, and H. Buschke. Abnormality of gait as a predictor of non-alzheimer's dementia. *New England Journal of Medicine*, 347:1761–1768, November 2002. 190

[169] L. M. Allan, C. G. Ballard, D. J. Burn, and R. A. Kenny. Prevalence and severity of gait disorders in alzheimer's and non-alzheimer's dementias. *Journal of the American Geriatrics Society*, 53(10):1681–1687, October 2005. 190

[170] T. Nakamura, K. Meguro, and H. Sasaki. Relationship between falls and stride length variability in senile dementia of the alzheimer type. *Gerontology*, 42(2):108–113, 1996. 190

[171] P. L. Sheridan and J. M. Hausdorff. The role of higher-level cognitive function in gait: executive dysfunction contributes to fall risk in alzheimer's disease. *Dementia and Geriatric Cognitive Disorders*, 24:125–137, 2007. 190

[172] N. Samudin, W. N. M. Isa, T. H. Maul, and W. K. Lai. Analysis of gait features between loaded and normal gait. pages 172–179, Morocco, November 2009. Int. Conf. on Signal-Image Technology and Internet-Based Systems. 190

[173] N. Rohila, B. Kumar, and N. Chauhan. Abnormal gait recognition. *Int. Journal on Computer Science and Engineering*, 2(5):1544–1551, 2010. 190

[174] R. Dzikansky, M. Kleiman, and G. Slater. *Chapter 13: Detection. What makes a suicide bomber seem suspicious?, in Terrorist suicide bombings*, pages 171–186. CRC Press, 2011. 190

[175] A. Sunderland. Balance and gait stability following sports-related concussion. *Australian Medical Student Journal*, 2(2):39–41, 2011. 190

[176] Y. Wahab and N. A. Bakar. Gait analysis measurement for sport application based on ultrasonic system. pages 20–24, Singapore, June 2011. IEEE Int. Symposium on Consumer Electronics. 190

[177] P. Langer. Video gait analysis in the podiatric sports medicine clinic. *American Academy of Podiatric Sports Medicine*, pages 149–154, June 2011. 190

[178] J. Lu and Y. P. Tan. Ordinary preserving manifold analysis for human age estimation. pages 90–95, United States, June 2010. IEEE Conf. on Computer Vision and Pattern Recognition Workshops. 190

[179] J. Lu and Y. P. Tan. Gait-based human age estimation. *IEEE Tran. on Information Forensics and Security*, 5(4):761–770, December 2010. 190

[180] Y. Makihara, M. Okumura, H. Iwama, and Y. Yagi. Gait-based age estimation using a whole-generation gait database. pages 1–6, United States, October 2011. Int. Joint Conf. on Biometrics. 190

[181] P. C. Chang, M. C. Tien, J. L. Wu, and C. S. Hu. Real-time gender classification from human gait for arbitrary view angles. pages 88–95, United States, December 2009. IEEE Int. Symposium on Multimedia. 190

[182] M. Hu, Y. Wang, Z. Zhang, and D. Zhang. Gait-based gender classification using mixed conditional random field. *IEEE Tran. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 41(5):1429–1439, October 2011. 190

[183] J. Lu, G. Wang, and T. S. Huang. Gait-based gender classification in unconstrained environments. Japan, November 2012. Int. Conf. on Pattern Recognition. 190

[184] J. Woo, S. C. Ho, J. Lau, S. G. Chan, and Y. K. Yuen. Age-associated gait changes in the elderly: pathological or physiological? *Neuroepidemiology*, 14:65–71, 1995. 192

[185] A. Gabell and U. S. Nayak. The effect of age on variability in gait. *Journal on Gerontol*, 39(6):662–666, November 1984. 192

[186] L. S. C. Raptopoulos, M. S. Dutra, A. S. de Lima, and W. M. Bessa. Comparison between male and female gait modes part i: kinematics of the ankle, knee and hip joints. Brazil, November 2005. Int. Congress of Mechanical Engineering. 192