

國立交通大學

資訊科學與工程研究所

博士論文

利用物件修補之數位內容還原與修改技術
Video Content Recovery and Modification
by Object Inpainting

研究生：凌誌鴻

指導教授：廖弘源 教授

陳永昇 教授

中華民國 一百零一年 六月

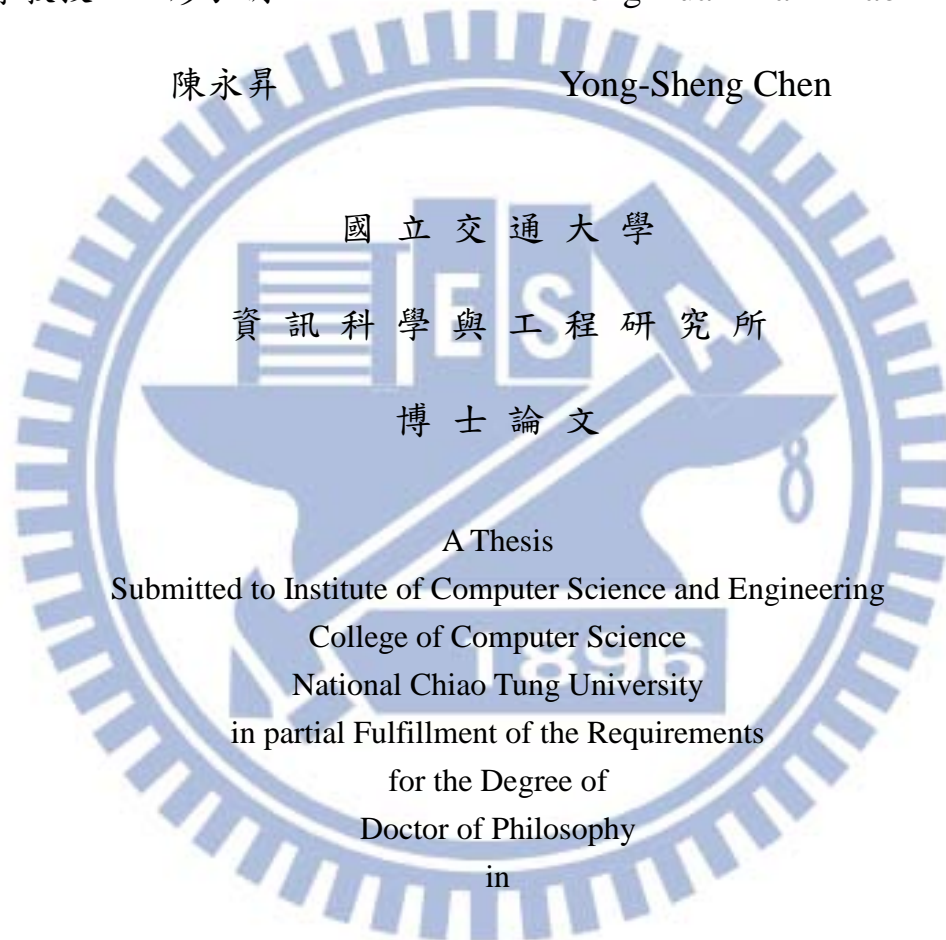
利用物件修補之數位內容還原與修改技術

Video Content Recovery and Modification by Object Inpainting

研究生： 凌誌鴻 Student： Chih-Hung Ling

指導教授： 廖弘源 Advisor： Hong-Yuan Mark Liao

陳永昇 Yong-Sheng Chen



國立交通大學

資訊科學與工程研究所

博士論文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer and Information Science

June 2012

Hsinchu, Taiwan, Republic of China

中華民國一百零一年六月

利用物件修補之數位內容還原與修改技術

學生：凌誌鴻

指導教授：廖弘源 博士

陳永昇 博士

國立交通大學資訊科學與工程研究所博士班

摘要

隨著數位攝影機的普及化，人們開始利用影像或影片記錄生活的點滴；因此，數位內容的還原及修改逐漸成為一個重要的研究議題。針對數位內容的還原，影片修補技術(video inpainting)可以自動地修補影片中內容缺失的部分，由於現存的影片修補技術對於影片中移動物體的修補成效不彰，因此在本論文中，我們提出兩種物件修補技術來修補影片中移動的物體；針對數位內容的修改，影片超解析度技術(video super-resolution)可以自動地增加影片在空間軸及時間軸上的解析度，由於現存的影片超解析度技術對於擴充影片中移動物體在時間軸上的解析度成效不彰，因此在本論文中，我們提出一種視訊內容擴充技術用來增加影片的畫面數同時擴充移動物體的動作內容。

在第一項研究中，我們先利用維度轉換將單張畫面上的物體資訊轉換成時空切片(spatio-temporal slice)上的物體軌跡資訊，每條軌跡紀錄物體某個部位沿著時間軸的變化趨勢，接著我們利用影像修補技術來修補時空切片上軌跡缺失的區域，最後經過維度反轉換，在單張畫面上我們重建被遮蔽物體可能的輪廓及位置。在下個步驟，根據重建的物體輪廓，我們從可用的物體姿態(posture)中選取適合的姿態並利用它取代畫面中被遮蔽的物體；當無可用的姿態時，我們提出一種姿態合成技術合成所需的姿態。第一種方法的效率容易受物體運動方向影響，因此我們在第二個方法中提出一種不受限於物體運動方向的物件修補技術。

在第二項研究中，我們先利用流形學習(manifold learning)將影片中物體運動的資訊轉換成在流形空間(manifold space)中運動軌跡的資訊；根據軌跡在流形空間中的分佈情況，我們描述動作連續的特性並定義兩種動作預測策略，利用定義的策略，我們可以預測被遮蔽物體可能的姿態。接著我們結合提出的預測策略及雙向預測方法，對於每個被遮蔽的物體選出一些可能的姿態，最後利用馬可夫隨機場(Markov random field)來選出最適當的姿態。

在第三項研究中，針對畫面數較低的影片，我們提出一種視訊內容擴充技術。我們先利用流形學習將影片中物體運動的資訊換換成在

流形空間中運動軌跡的資訊。在步驟二中，我們先利用提出的運動資料對齊方法將不同的運動資訊對齊並排列至張量(tensor)中，接著利用張量分解(tensor decomposition)從訓練的影片中抽取動作的資訊，並結合原始影片的人物資訊重建原始影片在高畫面數情況下動作軌跡在流形空間中分佈的情形，最後利用接著利用研究二中提出的方法選出適當的姿態並插入影片中適當的位置。



Video Content Recovery and Modification by Object Inpainting

Student : Chih-Hung Ling

Advisors : Dr. Hong-Yuan Mark Liao

Dr. Yong-Sheng Chen

Institute of Computer Science and Engineering
National Chiao Tung University

Abstract

With the popularization of digital cameras, people use image or video to record some snapshots of daily life. Hence, video content recovery and modification has become a popular research field in recent years. For video content recovery, video inpainting is considered as one of the most important techniques that can be used to automatically recover the missing regions of videos. However, most video inpainting algorithms generate artifacts if the object to be inpainted is seriously occluded or its motion is not complicated. To avoid generating such artifacts, we propose two different kinds of object-based video inpainting schemes that can solve the above-mentioned spatial inconsistency problem and the temporal continuity problem simultaneously in this dissertation. As to video content modification, video super-resolution is considered as one of

important techniques that can be used to automatically increase spatial and temporal resolution of videos. However, existing super-resolution methods may fail to produce realistic and smooth results while dealing with sequences of human motion. Hence, we propose a learning-based approach which can increase the frame rate of video and also enrich the motion content of human motion.

In our first work, we present a novel framework for object completion in a video. We transform object in frames into object trajectory in spatio-temporal slices, and complete the partially damaged object trajectories in the 2-D slices. The completed slices are then combined to obtain a sequence of virtual contours of the damaged object. Next, a posture sequence retrieval technique is applied to retrieve the most similar sequence of object postures based on virtual contours. Finally, a synthetic posture generation scheme is proposed to reduce the effect of insufficient postures.

In our second work, we propose a human object inpainting scheme that divides the whole process into three steps: human posture synthesis, graphical model construction, and posture sequence estimation. Human posture synthesis is used to enrich the number of postures. Then, all

postures are projected into manifold space to build a graphical model of human motion. We also introduce two constraints to confine the local motion continuity property. Finally, we perform both forward and backward prediction to derive local optimal solutions and then apply the Markov Random Field model to compute an overall best solution.

In our third work, we propose a learning-based approach to increase the temporal resolution of human motion sequences. We summarize the proposed framework in the following steps: graphical model construction, motion trajectory reconstruction and posture sequence estimation. In the first step, each motion sequence is projected into manifold space and represented as a motion trajectory. Then, we apply tensor decomposition to decompose motion trajectories into orthogonal factors. After that, we combine the motion factor from training sequences with the person factor from the input sequence to reconstruct the motion trajectory for the input sequence. Finally, we use the reconstructed motion trajectory combined with object inpainting technique to generate the final result.

誌 謝

漫長的求學階段終於畫上句點，這段期間受到許多人的幫忙及照顧，在此向曾經幫助我的師長、家人以及朋友獻上我的最誠摯的感謝。

首先要先感謝廖弘源老師，感謝老師提供了一個很好的研究環境，讓學生可以無憂無慮的作研究，並同時以身作則教導我們做研究及做人做事的道理，少了老師的教導，學生是無法完成這篇博士論文。同時也要感謝林嘉文老師，感謝老師九年來的教導，老師不僅在研究上給我許多指導，在學期間也給我許多的鼓勵及建議，讓學生在遇到挫折時仍能堅持完成學業。感謝陳永昇老師在系上事務上給予的許多幫忙及建議。感謝許秋婷老師在忙碌的生活中能願意撥空指導我論文的內容及寫作的方法。在此並感謝百忙之中抽空指導我口試的蔡文祥老師、莊仁輝老師、孫永年老師、范國清老師、柳金章老師及賴尚宏老師，對於本論文的指導以及建議。

感謝在中研院的學長(志文、祐銘，敦裕、士韋、易聰、明昉、家棟、立威、育駿)、同學(殷盈、興源)、學弟妹(堯麟、俊緯)及助理(amy、亦雲)，謝謝你們多年來的幫忙，這些年有了你們，讓枯燥的研究生生活變得更為有趣，讓徬徨無助的研究生生活中多了盞明燈。

最感謝我的父母，這麼多年來的支持，也感謝我的妹妹這幾年來

幫我擔負著照顧父母的責任，讓我可以完成我的學業，僅以此篇論文
來表達我對家人的萬分感謝。

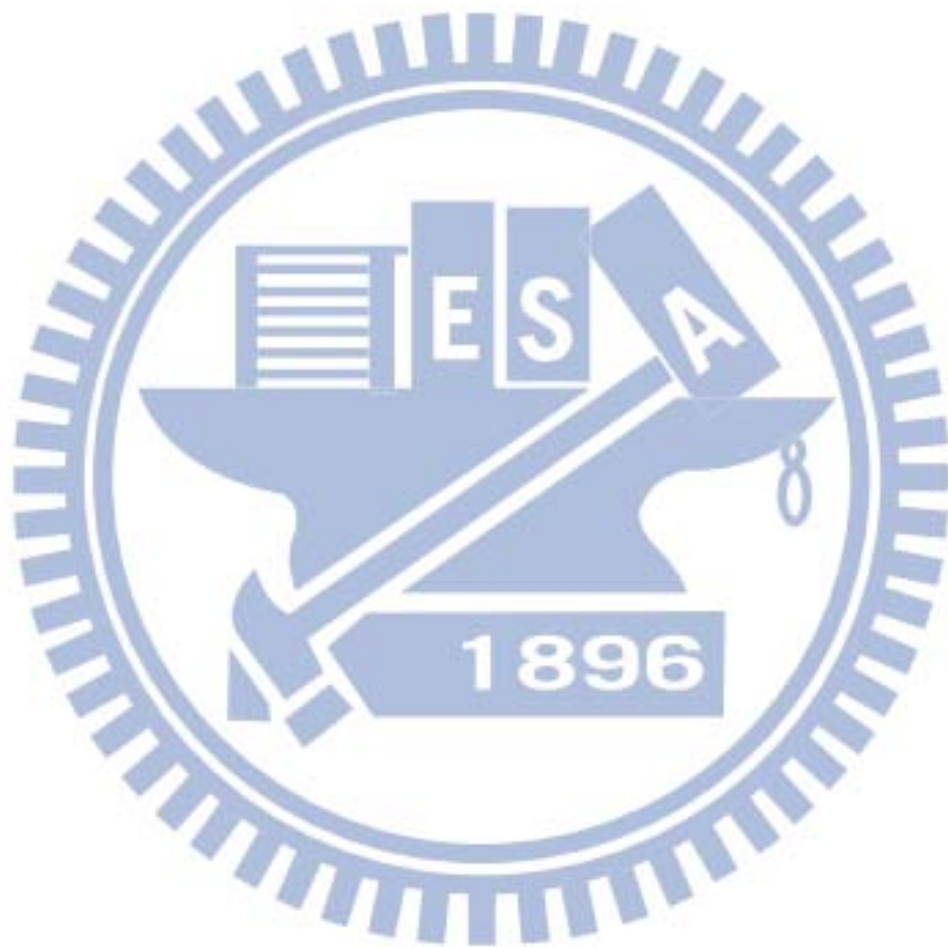


Table of Contents

摘要.....	I
Abstract.....	IV
誌謝.....	VII
Table of Contents.....	IX
List of Table.....	XII
List of Figure.....	XIII
1 Introduction.....	1
1.1 Motivation.....	1
1.2 Related Work.....	4
1.3 Overview of the Proposed Methods.....	7
1.4 Dissertation Organization.....	11
2 Virtual Contour Guided Video Object Inpainting Using Posture Mapping and Retrieval.....	12
2.1 Introduction.....	12
2.2 Occluded Object Completion Using Posture Sequence Matching.....	14
2.2.1 Overview.....	14
2.2.2 The Shape Context Descriptor.....	18
2.2.3 Virtual Contour Construction Using Spatio-Temporal Slices.....	20
2.2.4 Key Posture-based Posture Sequence Matching.....	28
2.2.5 Synthetic Posture Generation.....	32
2.3 Experimental Results.....	38

2.4	Summary	47
3	Human Object Inpainting Using Manifold Learning-Based	
	Posture Sequence Estimation.....	49
3.1	Introduction.....	49
3.2	Human Object Inpainting Using Posture Sequence	
	Estimation.....	53
3.2.1	Human Posture Synthesis	53
3.2.2	Graphical Model Construction.....	53
3.2.3	Posture Sequence Estimation.....	56
3.3	Experimental Results	65
3.4	Summary	75
4	Object Posture Temporal Super-Resolution Using Tensor	
	Decomposition-Based Manifold Learning	76
4.1	Introduction.....	76
4.2	Object Posture Temporal Super-Resolution.....	79
4.2.1	Overview of the Proposed Method.....	79
4.2.2	Graphical Representation of Object Motion.....	83
4.2.3	Temporal Super-Resolution Using Tensor	
	Decomposition–Based Manifold Learning.....	83
4.2.4	Posture Selection.....	89
4.3	Experimental Results	90
4.4	Summary	102
5	Conclusions and Future Work.....	103
5.1	Conclusions.....	103
5.2	Future Work	106
	Reference	107

Publication List117



List of Table

2.1	Run-time analysis of key operations in the proposed method.....	46
3.1.	Detailed information derived during the forward-backward prediction process	61
3.2	Comparison of the ground-truth postures and the reconstructed missing postures (The parts in black, red and gray represent the ground-truth postures, reconstructed postures, and perfectly matched portions, respectively)	69
3.3	Comparison of the ground-truth postures and the reconstructed missing postures (The parts in black, red and gray represent the ground-truth postures, reconstructed postures, and perfectly matched portions, respectively)	74
4.1	Comparison of the ground-truth postures and the up-sampled postures obtained by different methods for test sequence #1	93
4.2	Comparison of the ground-truth postures and the up-sampled postures obtained by different methods for test sequence #2.....	95
4.3	Comparison of the ground-truth postures and the up-sampled postures obtained by different methods for test sequence #3.....	98

List of Figure

2.1	Simplified flowchart of the proposed video inpainting scheme.....	15
2.2	Flowchart of the proposed object completion scheme.	17
2.3	Extracting the local context of a posture: (a) the object's original posture; (b) the object's silhouette described by a set of feature points; (c) the local histogram of a significant feature point, (d) extracting significant feature points of the object's silhouette using a convex hull surrounding the silhouette; and (e) the resultant significant feature points of the object's silhouette.....	19
2.4	Sampling a 3-D video volume comprised of several consecutive frames: (a) the original frame; (b) the object trajectory on a sampled XT plane s , indicated by the green lines in (a); (c) the original frame; (d) the object's trajectory on a sampled YT plane, indicated by the red lines in (c); (e) 2-D spatio-temporal slices sampled on a video shot, where the object's size varies due to non-pure horizontal motion; and (f) the removed occluded object trajectories on the XT plane sampled on the 2-D plane.	24
2.5	The notations used for the data and confidence terms in patch-based image inpainting [14].....	25
2.6	Virtual contours constructed by combining 2-D spatio-temporal slices derived via the patch-based inpainting	

method proposed in [14]. The left-hand side shows the virtual contours obtained by combining completed spatio-temporal slices without corrections, and the right-hand side shows the virtual contours with corrections.27

2.7 The process for converting available postures and virtual contours into a sequence of key posture labels. The blue frames and numbers indicate the frames with available postures and their corresponding key-posture labels. The orange frames and numbers indicate the frames with constructed virtual contours and their corresponding key-posture labels.32

2.8 Examples of using substring matching to solve the posture mapping problem. The length of the substring is 4. The blue numbers indicate the key-posture labels of available postures; the brown numbers indicate the labels of virtual contours; and the red numbers indicate the labels of available postures used to replace the occluded objects. In the first posture mapping, the available postures in frames 5, 6, $n-5$ and $n-4$ are deemed the best matches to replace the damaged objects in frames i , $i+1$, $j-1$ and j respectively. In the second mapping, however, a good match cannot be found for the damaged object in frame $i+2$ (with the virtual contour labeled “V”).....32

2.9 Synthesizing a new posture using available postures. The new posture is comprised of three components (the head, body, and legs) taken from different postures.34

2.10 Flowchart of the proposed synthetic posture generation

process.....	35
2.11 The constituent components of a posture are partitioned based on local variance extraction. The dashed lines which separate postures into constituent components are determined based on the distribution of local variance shown on the right-hand side.....	36
2.12 Test sequence #1 containing a single pedestrian: (a) some snapshots of the original video (ground-truths); (b) the virtual contours (on the left), which are constructed by combining the completed spatio-temporal slices and their corresponding best-match postures (on the right); (c) the corresponding completed frames; (d) comparison of the completed objects (on the left) and the ground-truths (on the right).....	40
2.13 Test sequence #2 with two people walking toward each other: (a) original video frames; (b) the virtual contours (on the left), which are constructed by combining the completed spatio-temporal slices and the corresponding best-match postures (on the right); (c) the completed frames (on the left) using the original key-postures and the additional synthetic postures and the corresponding frames composed from the completed 2-D slices (on the right).	42
2.14 Test sequence #3 containing two people walking toward each other (with a long occlusion period): (a) original video frames; (b) the virtual contours (on the left) and the corresponding best-match postures (on the right) without including synthetic postures; (c) the virtual contours (on the left) and the	

corresponding best-match postures (on the right) with the additional synthetic key-postures; and (d) the completed frames (on the left) using the original key-postures and the additional synthetic postures and the corresponding frames composed from the completed 2-D slices (on the right).	44
2.15 Test sequence #4: (a) some snapshots of the original video; (b) the corresponding best-match postures; and (c) the result derived by the proposed method.....	46
3.1 A graphical model of an object’s motion in a low-dimension manifold. The blue points represent the feature points of the postures, and the red lines connect two feature points whose corresponding postures appear in adjacent frames. In this example, occlusion occurs between frames i and j , so we try to find a motion path with l internal points that can be used to link points x_i and x_j	54
3.2 The neighborhood constraint.....	57
3.3 The motion tendency constraint.....	58
3.4 Some snapshots extracted from test sequence #1.....	60
3.5 (a)–(b) some forward prediction steps, (c)–(d) some backward prediction steps, and (e) the combined results of two-way prediction at time t	61
3.6 An example of the MRF process.....	64
3.7 The experiments on test sequence #1: (a) partial sequence of test sequence #1 in which the red rectangle indicates missing frames; (b) frames reconstructed by Ding <i>et al.</i> ’s approach; (c) frames reconstructed by Xu <i>et al.</i> ’s approach; (d) frames	

reconstructed by the proposed approach; and (e) the corresponding trajectory information of predicted object motion generated by the three approaches.	69
3.8 The experiments on test sequence #2: (a) some snapshots of the occluded object in the test sequence; (b) frames reconstructed by Ding <i>et al.</i> 's approach; (c) frames reconstructed by Xu <i>et al.</i> 's approach; (d) frames reconstructed by the proposed approach; and (e) the inpainting result derived by our approach.	71
3.9 The experiments on test sequence #3: (a) partial sequence of the test sequence in which the red rectangle indicates the 7 missing frames; (b) the frames reconstructed by Ding <i>et al.</i> 's approach; (c) the frames reconstructed by Xu <i>et al.</i> 's approach; (d) the frames reconstructed by the proposed approach; and (e) the corresponding trajectory information of predicted object motion generated by the compared approaches.....	74
4.1 Flowchart of the proposed posture super-resolution scheme	82
4.2 Illustration of tensor decomposition and arrangement: (a) a tensor data is decomposed into the product of core tensor and orthogonal factors, and (b) a tensor is flattened in two different ways to obtain flattened matrices.....	85
4.3 Illustration of the low-dimensional manifolds of two different posture sequences and the corresponding postures at the crests and troughs of the manifold.....	86
4.4 (a) The coordinates of the k postures of the LR input sequence. (b) We try to find k reference points among m reference points	

along the mean motion curve of all the HR learning sequences.
The index of the k reference points indicates the suitable
position in tensor of the input sequence postures.....88

4.5 Our scheme of arranging training postures into tensor data,
where the green rectangles represent unknown object postures
in the tensor. In tensor decomposition, we extract the motion
factor only from the training sequences as indicated by the red
rectangles and the person factor from the columns with
complete postures as indicated by the blue rectangles.89

4.6 Comparison of reconstruction accuracies with respect to the
ground-truth sequence with nine subsampling rates for test
sequence #1. The five compared methods include Xu *et al's*
approach [39], Ding *et al's* approach [10], Makihara *et al's*
approach [59], object inpainting [60] and the proposed
temporal SR approach.....101

Chapter 1

Introduction

1.1 Motivation

With the popularization of digital cameras, video content recovery and modification has become a popular research field in recent years. For video content recovery, video inpainting [1]-[11] has attracted a great deal of attention in recent years because of its powerful ability to fix/restore damaged videos and the flexibility it provides for editing home videos. It also ensures visual privacy in security applications [12]. More specifically, inpainting techniques have been used extensively for fixing/restoring damaged digital images [13]-[18]. Depending on how they restore damaged images, the techniques can be categorized into three groups: texture synthesis-based methods [13][14], partial difference equation-based (PDE-based) methods [15], and patch-based methods [16]. The concept of texture synthesis is borrowed from computer graphics. Its main purpose is to insert a chosen input texture into a damaged/missing region. In contrast, PDE-based approaches propagate information from the boundary of a missing region toward the center of

that region. They are suitable for completing a damaged image in which thin regions are missing. Texture synthesis and PDE-based propagation cannot handle cases of general image inpainting because the former does not consider structural information and the latter frequently introduces blurring artifacts. A patch-based approach [16], on the other hand, is much more suitable for image inpainting because it can produce high-quality visual effects and maintain the consistency of local structures. Because of the success of patch-based image inpainting, researchers have applied a similar concept in video inpainting; however, the issues that need to be addressed in video inpainting are much more challenging. Although video inpainting is a relatively new research area, a number of methods have been proposed in recent years. Generally, the methods can be classified into two types: patch-based methods [1]-[6], and object-based methods [7][8]. Patch-based methods often have difficulty handling spatial consistency and temporal continuity problems. In addition, patch-based approaches often generate inpainting errors in the foreground. As a result, many researchers have focused on object-based approaches, which usually generate high-quality visual results. Even so, some difficult issues still need to be addressed; for example, the artifacts

generated by inpainting completely occluded object or inpainting occluded object with non-periodic motion. Hence, in this dissertation, we propose two different kinds of object-based video inpainting schemes that can solve the spatial inconsistency problem and the temporal continuity problem simultaneously.

As to video content modification, super resolution-based (SR-based) methods have attracted much attention for their ability in enhancing the spatial or temporal resolution of low-resolution (LR) images/videos [48]–[54]. While dealing with sequences of human motion, existing SR-based methods may fail to produce realistic and smooth results if no special efforts are taken to handle the non-rigid human motion. Since human motion usually contains repeated postures, one may insert interpolated postures into the LR input sequence to increase the temporal resolution. In order to generate postures and animate animal/human motion, Xu *et al.* [39] proposed to animate motions by minimizing a predefined energy function. Since the energy minimization process did not include a human motion model, the performance is unstable and very sensitive to the selected parameters. Therefore, some existing methods [10] [59] develop their approach under the constraint of periodic motion.

To overcome the above mentioned drawbacks, we propose the use of learning-based approach to extract motion tendency from a set of learning sequences and then synthesize human motion using the learned motion tendency as the prior information.

1.2 Related Work

Conventional video inpainting methods can be roughly classified into two types: the first type is patch-based [1]-[6] and the other type is template-based [7][8]. In [1], Patwardhan *et al.* proposed a video inpainting technique that makes use of motion information and image inpainting technique together. Motion information is adopted to help find the most suitable patch. In [2], the space-time volume is sliced up into motion manifolds to perform video completion. The proposed manifolds are composed of two-dimensional patches (one for the spatial dimension and the other for the temporal dimension). These patches cover the entire trajectory of pixels, and the method in [2] applies Sun *et al.*'s approach [17] to inpaint those missing regions. However, these approaches would cause spatial or temporal structure inconsistency artifacts. In [4], Wexler *et al.* adopted a 3-D fix-sized patch as a unit for video inpainting. The

value of a missing pixel is estimated by a set of constituent patches and a multiscale solution is used to speed up the process. In [5], Cheung *et al.* introduced a probabilistic patch model for video inpainting. They use a video epitome method to compress an original video by learning, after that the epitome is used to synthesize data for the damaged areas of a video.

In the template-based video inpainting category, Cheung *et al.* [7] proposed a technique to deal with the problem of missing objects in videos captured by a stationary camera. All available object templates are used to inpaint the foreground. Then, for each missing object, a fixed-size sliding window that covers the missing object and its neighboring templates is used to find the most similar object template. Although the sliding window can help find similar object templates, the inpainting result may be unsatisfactory if the number of postures is insufficient. Furthermore, a good filling position is crucial for an object inpainting process because an inappropriate position may cause visually annoying artifacts. In [8], Jia *et al.* proposed a user-assisted video layer segmentation technique that decomposes an input video into color and illumination videos. A tensor voting technique is then used to address the

pertinent spatio-temporal issues in background and foreground. Image repairing is used for background inpainting and occluded objects are reconstructed by synthesizing other available objects. However, a synthesized object created under this approach does not have a real trajectory, so the approach is only suitable for objects with periodic motion.

As to human motion animation, Ding *et al.* [10] proposed a rank minimization approach to model and synthesize human motion for video inpainting. They first projected the observed data into a low-dimension manifold and then organized the embedded features to form a Hankel matrix. The missing features in the Hankel matrix are determined by minimizing the rank of Hankel matrix. Finally, they applied the Radial Basis Function (RBF) to inversely transform the embedded features back to the observation domain. This rank minimization approach would usually produce good results as far as the object's motion is periodic. Makihara *et al.* [59] proposed a reconstruction-based method to synthesize periodic human motion with high frame rate from a single periodic motion sequence. The human motion data are first transformed into embedded features in a low-dimension manifold. Then, they

iteratively conducted phase registration and motion trajectory reconstruction within an energy minimization process. Under the constraint of periodic motion, their method could also produce good experiment results.

1.3 Overview of the Proposed Methods

Our literature survey shows that most video inpainting algorithms generate artifacts if the object to be inpainted is completely occluded or its motion is not periodic. To void generating such artifacts, a posture sequence estimation process of good accuracy is required for object inpainting. In this dissertation, we propose two different kinds of object-based video inpainting schemes that can solve the spatial inconsistency problem and the temporal continuity problem simultaneously. As to human motion animation, some kinds of method [10] [59] have performance limitation of periodic motion. Therefore, in this dissertation, we propose to extract motion tendency form a set of learning sequences as prior information and then synthesize human motion using the extracted motion tendency.

Virtual Contour Guided Video Object Inpainting Using Posture Mapping and Retrieval

In this work, we present a novel framework for object completion in a video. To complete an occluded object, our method first samples a 3-D volume of the video into directional spatio-temporal slices, and performs patch-based image inpainting to complete the partially damaged object trajectories in the 2-D slices. The completed slices are then combined to obtain a sequence of virtual contours of the damaged object. Next, a posture sequence retrieval technique is applied to the virtual contours to retrieve the most similar sequence of object postures in the available non-occluded postures. Key-posture selection and indexing are used to reduce the complexity of posture sequence retrieval. We also propose a synthetic posture generation scheme that enriches the collection of postures so as to reduce the effect of insufficient postures. The experiment results demonstrate that the proposed method can maintain the spatial consistency and temporal motion continuity of an object simultaneously.

Human Object Inpainting Using Manifold Learning-Based Posture

Sequence Estimation

In this work, we propose a human object inpainting scheme that divides the process into three steps: human posture synthesis, graphical model construction, and posture sequence estimation. Human posture synthesis is used to enrich the number of postures in the database, after which all the postures are used to build a graphical model that can estimate the motion tendency of an object. We also introduce two constraints to confine the motion continuity property. The first constraint limits the maximum search distance if a trajectory in the graphical model is discontinuous; and the second confines the search direction in order to maintain the tendency of an object's motion. We perform both forward and backward prediction to derive local optimal solutions. Then, to compute an overall best solution, we apply the Markov Random Field model and take the potential trajectory with the maximum total probability as the final result. The proposed posture sequence estimation model can help identify a set of suitable postures from the posture database to restore damaged/missing postures. It can also make a reconstructed motion sequence look continuous.

Object Posture Super-Resolution Using Tensor Decomposition-Based Manifold Learning

In this work, we propose a learning-based approach to increase the temporal resolutions of human motion sequences. Given a set of high resolution motion sequences, our idea is first to learn the motion tendency from this learning dataset and then synthesize new postures for the low-resolution sequence according to the learned motion tendency. To ensure the synthesized motion should preserve the learned motion tendency as well as its personal characteristic, we propose using tensor decomposition to decompose motion data into two orthogonal factors. We summarize the proposed framework in the following steps: (1) Each motion sequence is first projected into a low-dimension manifold space, where the local distance between postures could be better preserved. We then represent each of the projected motion sequences as a motion trajectory, and conduct tensor decomposition on the motion trajectories to extract the two orthogonal factors: motion and person. (2) We combine the motion factor from training sequences with the person factor from the input sequence to reconstruct the motion trajectory for the input sequence. (3) We use the reconstructed motion trajectory combined with object

inpainting technique to generate the final result. Our experimental results demonstrate the effectiveness of the proposed method, and also show its outperformance over two existing approaches.

1.4 Dissertation Organization

The remainder of this dissertation is organized as follows. In Chapter 2, the proposed framework for virtual contour guided video object inpainting using posture mapping and retrieval is described in detail. In Chapter 3, the proposed framework for human object inpainting using manifold learning-based posture sequence estimation is described in detail. In Chapter 4, the proposed object posture super-resolution using tensor decomposition-based manifold learning is described in detail. Finally, in Chapter 5, we draw our conclusions and future work.

Chapter 2

Virtual Contour Guided Video Object Inpainting Using Posture Mapping and Retrieval

In this Chapter, we describe the proposed framework for virtual contour guided video object inpainting using posture mapping and retrieval. First, we give an introduction about this research topic. The proposed approach is then described. Next, we detail the experiment results. Finally, we present our conclusions.

2.1 Introduction

Video inpainting [1]-[11] has been a very popular research topic recently due to its powerful ability to fix/restore damaged videos and the flexibility it provides for editing home videos. Researchers working in this field divide video inpainting methods into patch-based methods [1]-[6] and object-based methods [7][8]. A patch-based method often has difficulty handling spatial consistency and temporal continuity problems. As a result, many researchers have focused on object-based approaches, which usually generate high-quality visual results. Even so, some difficult issues still need to be addressed; for example, the unrealistic trajectory

problem and the inaccurate representation problem caused by an insufficient number of postures in the database. In order to solve these problems, we propose an object-based video inpainting scheme. The scheme is comprised of three steps: virtual contour construction, key-posture selection and mapping, and synthetic posture generation. The contribution of this work is three-fold. First, we propose a scheme that is able to derive the virtual contour of an occluded object. The contour provides a fairly precise initial estimate of the posture and filling location of the occluded object, even if the object is completely occluded. Therefore, the virtual contour is suitable for finding a good replacement for the occluded object from the available postures in the input video. Second, we propose a key posture-based mapping scheme that converts the posture sequence retrieval problem into a substring matching problem, thereby reducing the computational complexity significantly, while maintaining the matching accuracy. Since the occluded objects are completed for a whole sub-sequence rather than for individual frames, the temporal continuity of object motion is maintained as well. Third, for a sequence in which we cannot find a sufficiently rich set of available postures for completing occluded postures, our proposed synthetic

posture generation scheme can effectively enrich the database of postures by combining the constituent parts of different available postures. As a result, improved inpainting performance is achieved.

2.2 Occluded Object Completion Using Posture Sequence Matching

2.2.1 Overview

The proposed object-based video inpainting scheme can maintain the spatial consistency and temporal motion continuity of an object simultaneously. The scheme can also handle the problem of insufficiency of available postures. Figure 2.1 shows a block diagram of the proposed scheme. Initially, we assume that the objects to be removed and the occluded objects to be restored have been extracted by an automatic object segmentation scheme [19], or by an interactive extraction scheme [20]-[22]. After object extraction, the occluded objects and the background are completed separately. We also assume that the trajectory of each occluded object can be approximated by a linear line segment during the period of occlusion. This assumption is reasonable for many practical applications because the duration of an occlusion is typically short, and an object does not usually perform complex motions during

such a short period.

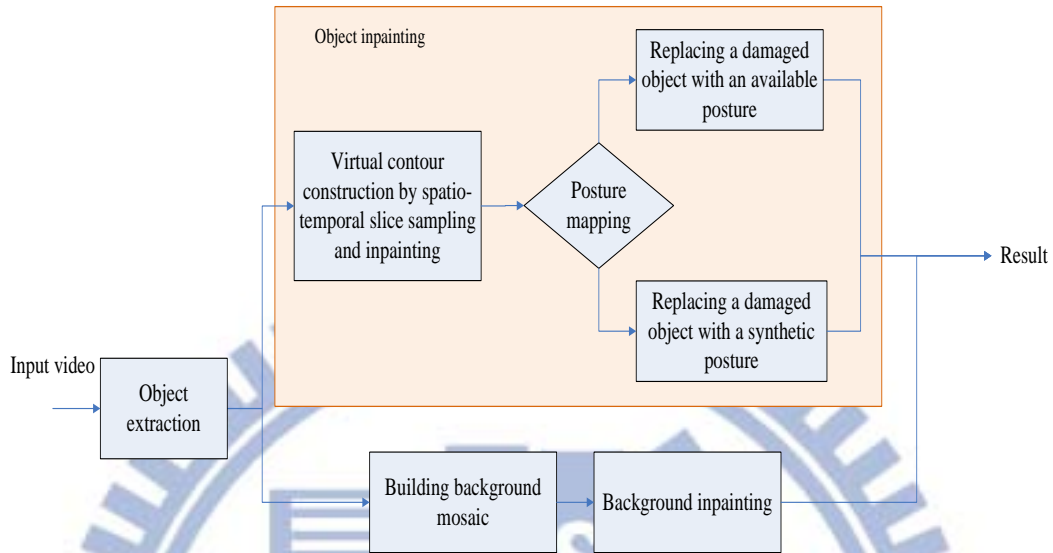


Figure 2.1 Simplified flowchart of the proposed video inpainting scheme.

Our primary goal is to solve the problem of completing partially or totally occluded objects in a video. Figure 2.2 shows the flowchart of the proposed object completion scheme which is comprised of three steps: virtual contour construction, key posture-based posture sequence matching, and synthetic key posture generation. The first step of object inpainting involves sampling a 3-D volume of video into directional spatio-temporal slices. Then a patch-based (exemplar-based) image inpainting [16] operation is performed to complete the partially damaged object trajectories in the 2-D spatio-temporal slices. The objective is to maintain the trajectories' temporal continuity. The completed

spatio-temporal slices are then combined to form a sequence of virtual contours of the target object to infer the missing part of the object's posture [29]. Next, the derived virtual contours and a posture sequence matching technique are used to retrieve the most similar sequence of object postures from among the available non-occluded postures. The available postures are collected from the non-occluded part of the input video. We perform key posture selection, indexing, and coding operations to convert the posture sequence retrieval problem into a substring search problem, which can be solved efficiently by existing substring-matching algorithms [23]. If a virtual contour cannot find a good match in the database of available postures, we construct synthetic postures by combining the constituent components of key postures to enrich the posture database. This process mitigates the problem of insufficient available postures. After retrieving the most similar posture sequence, the occluded objects are completed by replacing the damaged objects with the retrieved ones.

For background inpainting, we follow the background mosaics method proposed in [1]. The method first constructs a background mosaic for each video shot based on global motion estimation (GME), and then

finds the corresponding available data in the background mosaic for each pixel in a missing region. The data is used to fill the missing regions and thereby achieve spatio-temporal consistency in the completed background. Since background inpainting is not the focus of this work we do not consider its implementation in detail.

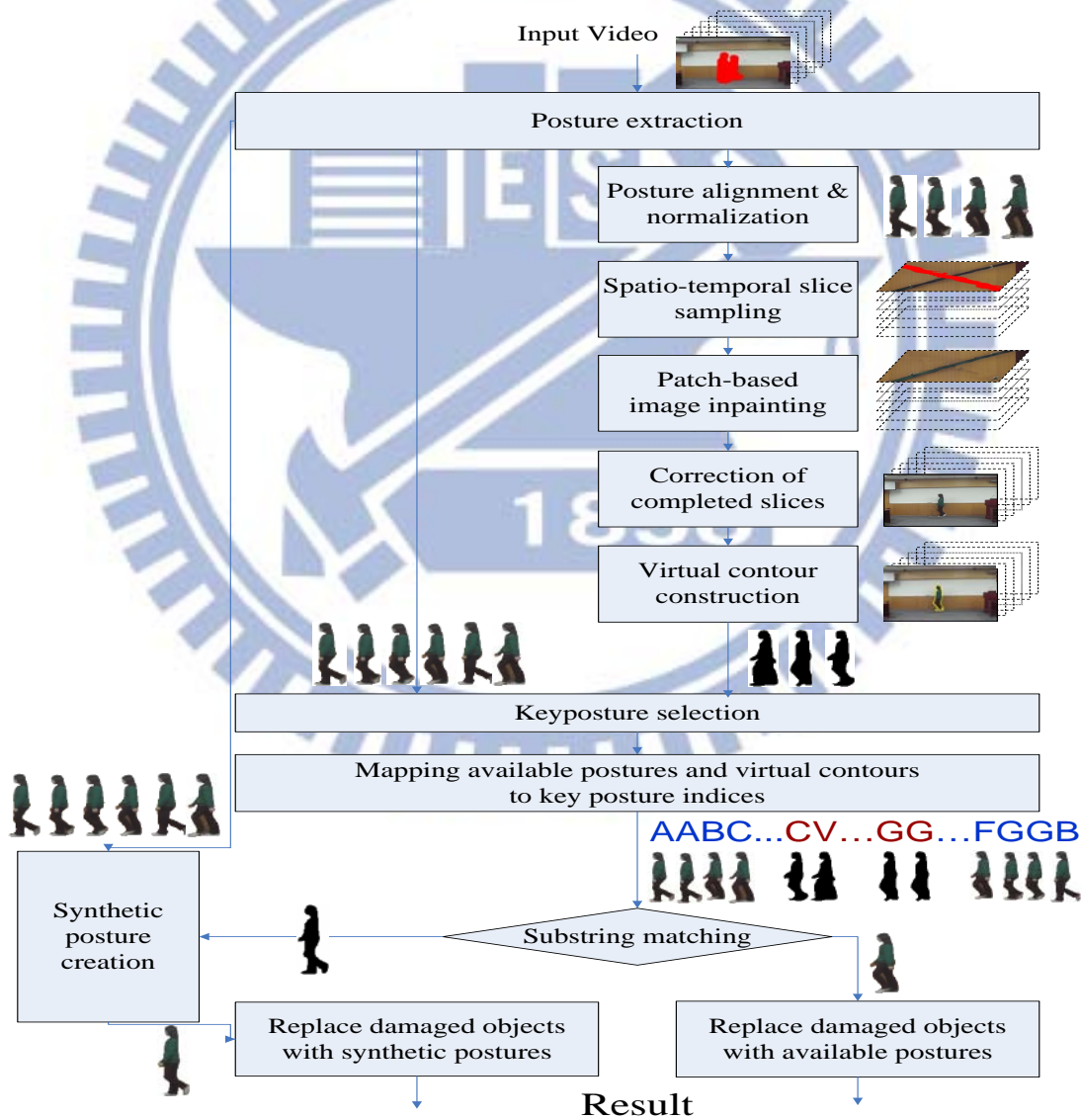


Figure 2.2 Flowchart of the proposed object completion scheme.

2.2.2 The Shape Context Descriptor

Before discussing the proposed method in detail, we describe the shape context descriptor in [23][24], which we use for posture alignment/normalization and key posture selection. The descriptor is invariant to translation, scaling, and rotation; and it is even robust against small amounts of geometrical distortion, occlusion and outliers. As shown in Figure 2.3, given an object image (Figure 2.3 (a)), the descriptor selects a set of feature points to describe the object's silhouette (Figure 2.3 (b)). The object's local shape context is described by the local histograms of the regions centered at the feature points. Under this method, for each feature point, a circle with radius r (Figure 2.3 (c)) is used to find the local histogram. The circle is then divided into N_{bin} partitions and the number of feature points in each partition is calculated, resulting in a histogram with N_{bin} bins. The value of N_{bin} is empirically set to be 60 for all sequences. The cost of matching two different sampled points which belong to two different postures can be defined as follows:

$$F(a_i, c_j) = \frac{1}{2} \sum_{k=1}^{N_{bin}} \frac{[h_{a_i}(k) - h_{c_j}(k)]^2}{h_{a_i}(k) + h_{c_j}(k)}, \quad (2.1)$$

where $h_{a_i}(k)$ and $h_{c_j}(k)$ denote the k -th bin of the two sampled points

a_i and c_j , respectively. The value of N_{bin} is empirically set to be 60 for all sequences, and the value of r is determined by an algorithm described in [24]. The best match between two different postures can be accomplished by minimizing the following total matching cost:

$$H(\pi) = \sum_j F(a_j, c_{\pi(j)}), \quad (2.2)$$

where π is a permutation of $1, 2, \dots, n$. Because of the one-to-one matching requirement, shape matching can be considered as an assignment problem that can be solved by a bipartite graph matching method. Therefore, the shape context distance between two shapes A and C can be computed by

$$F_{sc}(A, C) = \frac{1}{N_A} \sum_i F(a_i, c_{\pi(i)}) + \frac{1}{N_C} \sum_j F(a_j, c_{\pi(j)}), \quad (2.3)$$

where N_A and N_C are the numbers of sample points on the shape A and C , respectively.

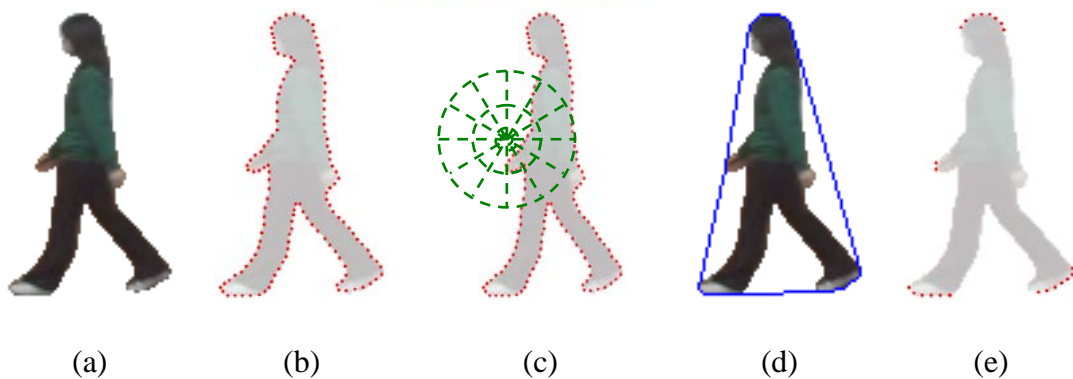


Figure 2.3 Extracting the local context of a posture: (a) the object's original

posture; (b) the object's silhouette described by a set of feature points; (c) the local histogram of a significant feature point, (d) extracting significant feature points of the object's silhouette using a convex hull surrounding the silhouette; and (e) the resultant significant feature points of the object's silhouette.

2.2.3 Virtual Contour Construction Using Spatio-Temporal Slices

The main difficulty in completing a damaged video object is that the information left in a badly damaged object is usually insufficient to reconstruct the object properly by using spatio-temporal clues. Furthermore, completing an object frame-by-frame often causes temporal discontinuity in the object's appearance and motion, since a frame-wise completion process does not consider an object's temporal dependency in consecutive frames. Such temporal discontinuity results in visually annoying artifacts like flickering and jerkiness. To ensure that a completed object is visually pleasing, it is important to extract a set of features from a damaged object in a number of consecutive frames. As a result, the features not only represent the object's characteristics (e.g., motion, appearance, and posture), but also take its temporal continuity into account.

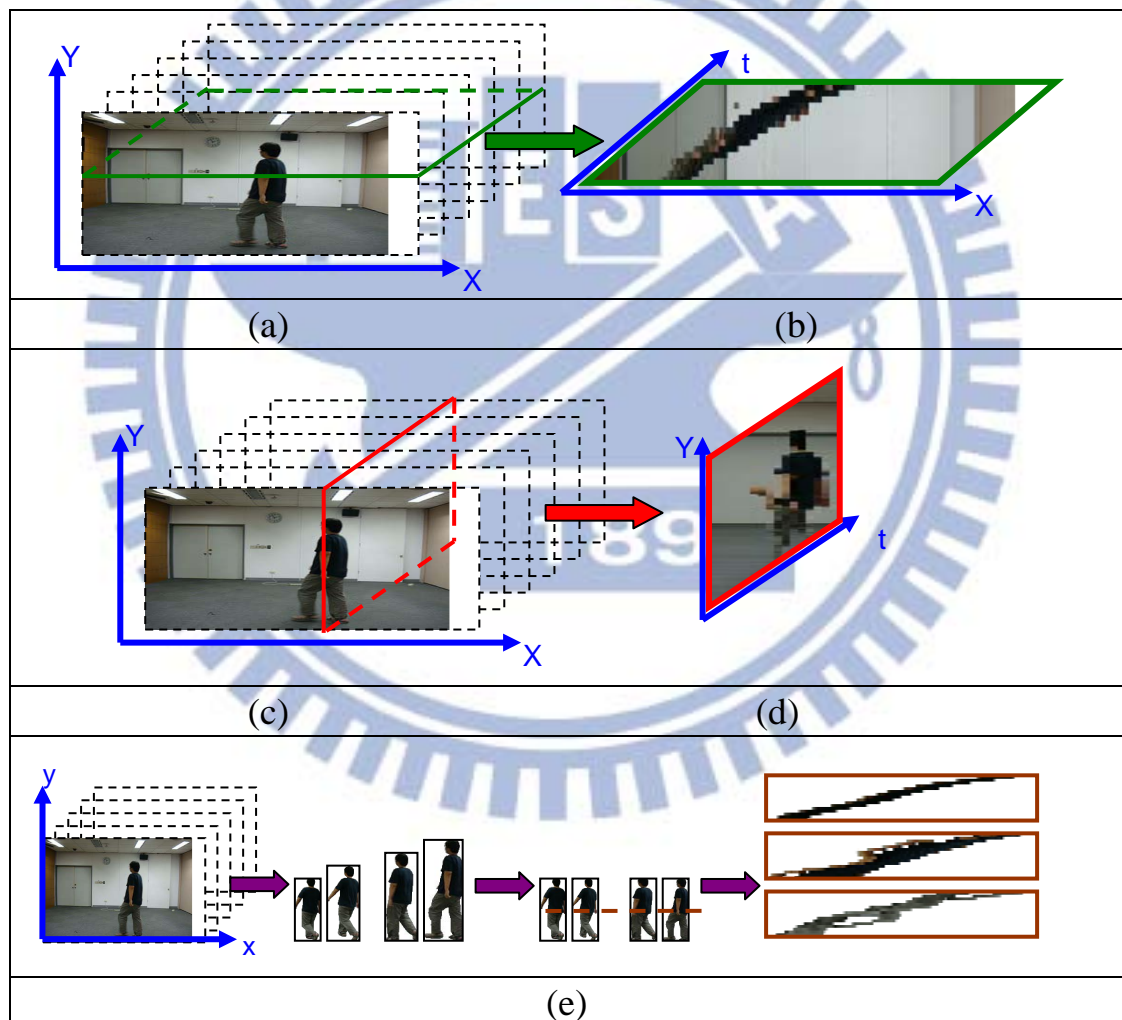
Manifold learning based methods [10][25] have been proposed to recover the damaged/missing poses of an occluded object. Although the

consecutive poses of an object with regular and cyclic motion can be well represented by a low-dimensional manifold embedded in a high-dimensional visual space, poses with non-regular motions (e.g., transitions in two types of motions) are usually not the case. As a result, mapping reconstructing a high-dimensional video object with irregular or non-cyclic motion from the object's low-dimensional manifold approximation usually leads to annoying artifacts (e.g., ghost images).

As mentioned earlier, we use spatio-temporal slices of a video to derive virtual object contours, which are then used as features to infer the occluded object poses. More specifically, after object extraction and removal, we sample a 3-D video volume comprised of several consecutive frames to obtain a set of directional 2-D spatio-temporal slices, as shown in Figure 2.4. For example, if a 3-D video volume (Figure 2.4 (a)) is sampled at different Y values (Figure 2.4 (b)), each resulting XT slice represents the horizontal trajectory of an object over time. The trajectory can fully capture an object's motion if it only has horizontal motions. Other directional sampling schemes can be used to deal with objects that have different motion directions. Note that a non-pure horizontal motion will cause an object's size to vary over time

due to the zoom-in/zoom-out effect, as shown in Figure 2.4(c). In this case, posture alignment and normalization can be used to avoid the inference of different posture scales. Without loss of generality, we use the largest posture of an object as a reference for aligning and normalizing the other postures. First, we establish the correspondence between the contour points of every two adjacent postures by shape matching [23][24]. The affine transformation parameters between the largest posture and the others can then be estimated from the corresponding points using the least squares optimization method. As a result, all postures are aligned and normalized with the largest posture via the affine transformations. As shown in Figure 2.4(d), after removing the foreground object and posture alignment, object occlusion results in incomplete trajectories of the object in the spatio-temporal slices. The missing regions of object trajectories in the 2-D spatio-temporal slices must be completed using an image inpainting method before composing a virtual contour. Because an object's occlusion period is usually short, we assume that the occluded part of a motion trajectory in a 2-D slice can be approximated by a line. Based on this assumption, the occluded part in

each directionally sampled slice can be inpainted well. Since the trajectory of an object on each 2-D slice records the locations of the same part of object over time, as long as the missing regions of trajectories are completed properly, the reconstructed trajectories will be continuous, thereby preserving the temporal continuity of an object.



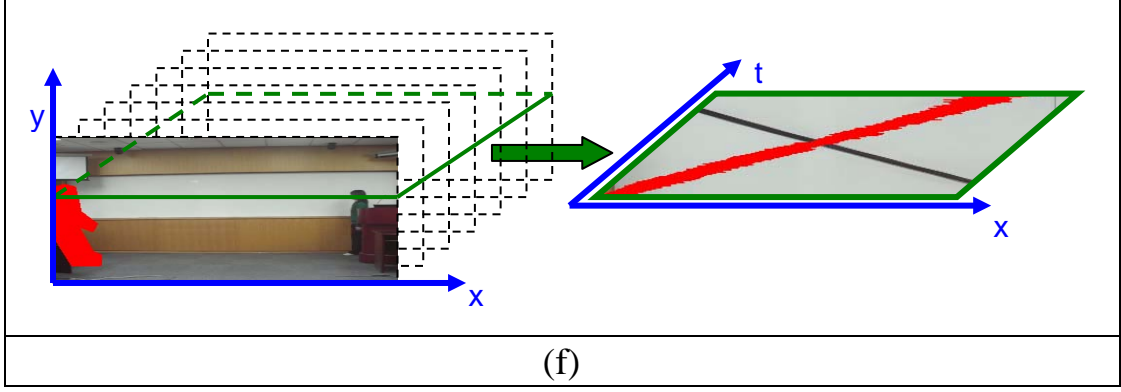


Figure 2.4 Sampling a 3-D video volume comprised of several consecutive frames: (a) the original frame; (b) the object trajectory on a sampled XT plane s , indicated by the green lines in (a); (c) the original frame; (d) the object's trajectory on a sampled YT plane, indicated by the red lines in (c); (e) 2-D spatio-temporal slices sampled on a video shot, where the object's size varies due to non-pure horizontal motion; and (f) the removed occluded object trajectories on the XT plane sampled on the 2-D plane.

To obtain continuous object trajectories, we use the patch-based image inpainting scheme proposed in [16] to complete missing regions in the spatio-temporal slices. The method first determines the filling order of the missing regions based on the confidence term and data term as follows:

$$P(p) = C(p) \cdot D(p), \quad (2.4)$$

where $P(p)$ represents the priority of a missing region p ; and $C(p)$ and $D(p)$ denote the confidence term and the data term expressed in (2.5) and (2.6) respectively.

$$C(p) = \frac{\sum_{q \in \Psi_p \cap (I - \Omega)} C(q)}{|\Psi_p|}, \quad (2.5)$$

$$D(p) = \frac{|\nabla I_p^\perp \cdot n_p|}{\alpha}, \quad (2.6)$$

where $|\Psi_p|$ represents the area of region Ψ_p , α is a normalization factor, n_p denotes the unit vector orthogonal to the front $\delta\Omega$ at point p , and \perp stands for the orthogonal operator, as illustrated in Figure 2.5.

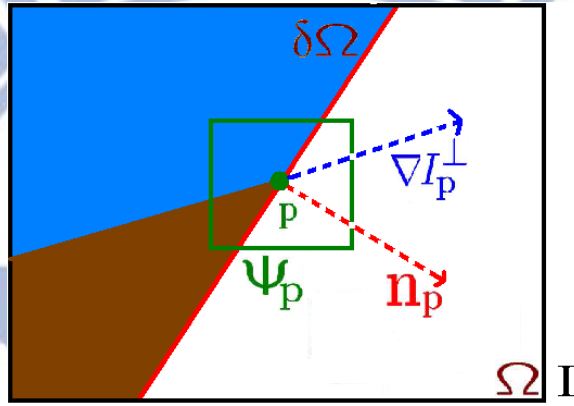


Figure 2.5 The notations used for the data and confidence terms in patch-based image inpainting [14].

Based on the filling order, a missing region is filled with the most similar neighboring patches (measured by the sum of squared differences). After completing each spatio-temporal slice of a video frame, we use the Sobel edge detector to find the boundary of the object's trajectory in the slice. Then, the completed spatio-temporal slices are combined to construct a virtual contour, which is used to guide the subsequent posture mapping and retrieval process.

Sometimes, image inpainting errors lead to imprecise virtual contours, making it difficult to retrieve correct postures for object inpainting. To resolve this problem, we use the object tracking scheme proposed in [27] to correct image inpainting errors. To inpaint an occluded object, our method tracks the object in the non-occlusion period to obtain their positions. Accordingly, each spatio-temporal slice is then divided into two regions, the background region and the foreground trajectory, which allows us to apply image inpainting to the regions separately and thereby avoid inpainting errors. That is, available foreground information will only be used to infill the missing region of foreground region, and vice versa. Figure 2.6 shows that the tracking-based correction technique significantly reduces the distortion of a virtual contour caused by inpainting errors.

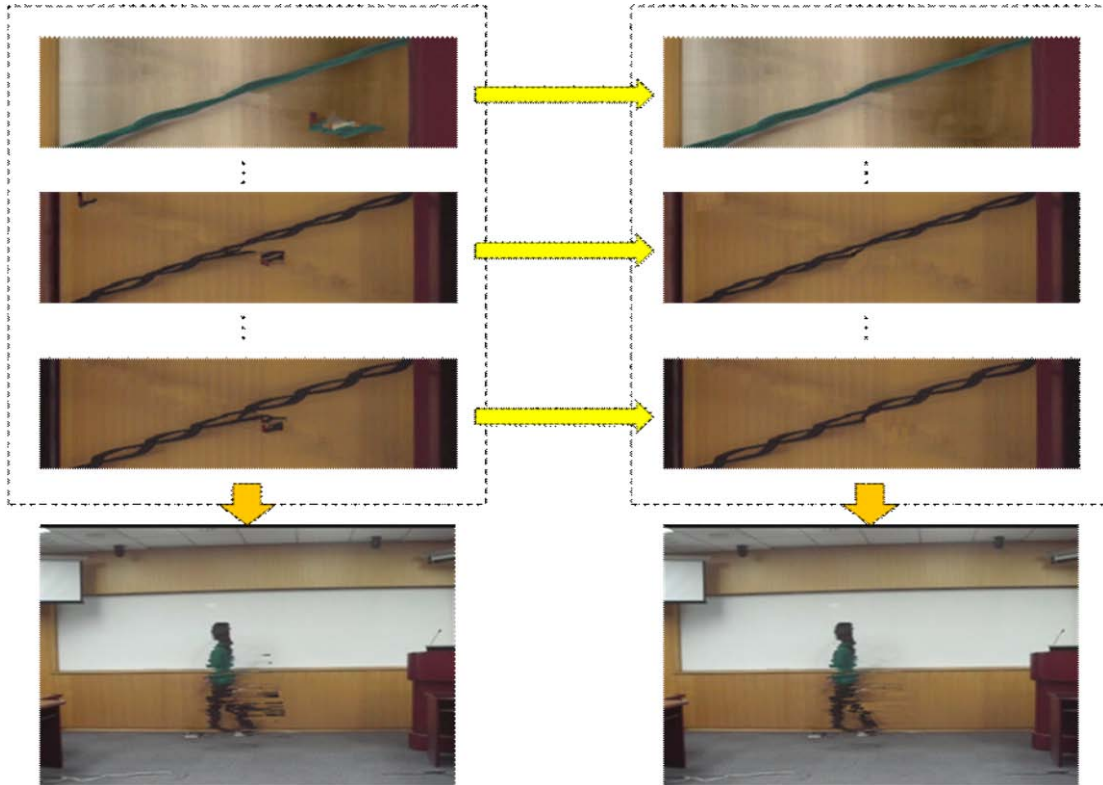


Figure 2.6 Virtual contours constructed by combining 2-D spatio-temporal slices derived via the patch-based inpainting method proposed in [14]. The left-hand side shows the virtual contours obtained by combining completed spatio-temporal slices without corrections, and the right-hand side shows the virtual contours with corrections.

The rationale behind the proposed virtual contour construction method is that if the continuity of object trajectories can be maintained in individually completed spatio-temporal slices, then the motion continuity of an object reconstructed by combining all the inpainted slices will also be maintained. Thus, so long as the linear line motion assumption holds during the occlusion period, a virtual contour can provide fairly precise information about the posture and filling location of an occluded object,

even if the object is badly damaged.

2.2.4 Key Posture-based Posture Sequence Matching

After composing a sequence of consecutive virtual contours, we use them to match the most similar posture sequence in the set of available postures to complete the occluded objects. To simplify the posture sequence matching process, we use the key posture selection method proposed in [24] to select the most representative postures from among the available postures. The method also uses the shape context descriptor in [24] to measure the similarity between two postures. As illustrated in Figure 2.3, given an object's posture (Figure 2.3 (a)), a set of feature points are selected to describe the object's silhouette (Figure 2.3 (b)). To reduce the complexity of posture matching without sacrificing the matching accuracy significantly, a convex hull bounding the silhouette (Figure 2.3 (d)) is used to select a subset of key feature points (Figure 2.3 (e)) to describe the shape context of the object. The similarity between two postures is evaluated by matching the two corresponding posture silhouettes by (2.3). A posture is deemed a key posture if its degree of similarity to all key postures exceeds a predefined threshold, $TH_{posture}$,

that is empirically set to be 0.08. The key-posture selection algorithm is summarized below.

Algorithm: Key Posture Selection

The set of key-postures $Q = \{q_1, q_2, \dots, q_n\}$

The available posture database $B = \{b_1, b_2, \dots, b_n\}$

For $i = 1$ to n

{

 If ($Q = \phi$)

$Q = Q \cup b_i$

 else if ($H(b_i, q_j) > TH_{\text{posture}}, \forall q_j$)

$Q = Q \cup b_i$

}

After the key posture selection process, each key posture is labeled with a unique number. The virtual contour of each available posture is then matched with the key posture that has the most similar context, as defined in (2.3). If a virtual contour cannot be matched in this way, it is given a special label. As a result, a sequence of contiguous available postures and virtual contours can be converted into a string of key-posture labels based on the temporal order, as shown in Figure 2.7.

After the encoding process, the problem of retrieving the most similar sequence of postures for a sequence of virtual contours becomes a substring matching problem [26] that, given an input segment of codes, searches for the most similar substring in a long string of codes. The occluded objects are then replaced with the retrieved sequence of available postures. Figure 2.8 shows two examples of using substring matching to solve the posture mapping problem. During the occlusion period, a string of labels in a fixed-size sliding window (the size is 4 in the example) is matched to the substring of labels in the normal periods. We use two sliding windows that respectively start from the two ends of the occlusion period and move toward the center of the period. Each sliding window overlaps with the neighboring normal period by half a window. As a result, half of the labels in the initial string are derived from available postures and the remaining labels are obtained from the virtual contours. As illustrated in the first example of Figure 2.8, the left sliding window initially consists of four postures encoded as “BBCC” including two available postures (the “BB” part) in frames $i-2$, $i-1$ and two virtual contours (the “CC” part) in frames i , $i+1$. The right sliding window initially contains four postures encoded as “EFGG” where “EF”

represents the two virtual contours in frame $j-1$ and j . and “GG” represents the two available postures in frames $j+1$, and $j+2$, respectively. In this example, the available postures in frames 5, 6, $n-5$ and $n-4$ of the two initial sliding windows are deemed the best-match sequence to replace the damaged objects in frames i , $i+1$, $j-1$ and j . In the second matching, however, a good match cannot be found for the damaged object in frame $i+2$ (with virtual contour label “V”) after substring matching. Our method handles such situations by constructing synthetic key-postures, as will be discussed later.

Using the proposed key-posture selection and mapping method to encode a sequence of virtual contours and available postures with a compact representation of key-posture labels has two advantages. First, since there are many efficient substring matching algorithms, converting the posture sequence retrieval problem into a substring matching problem reduces the computational complexity substantially. Second, as the occluded objects are completed for a whole sub-sequence rather than for individual frames, the temporal continuity of object motion is maintained.

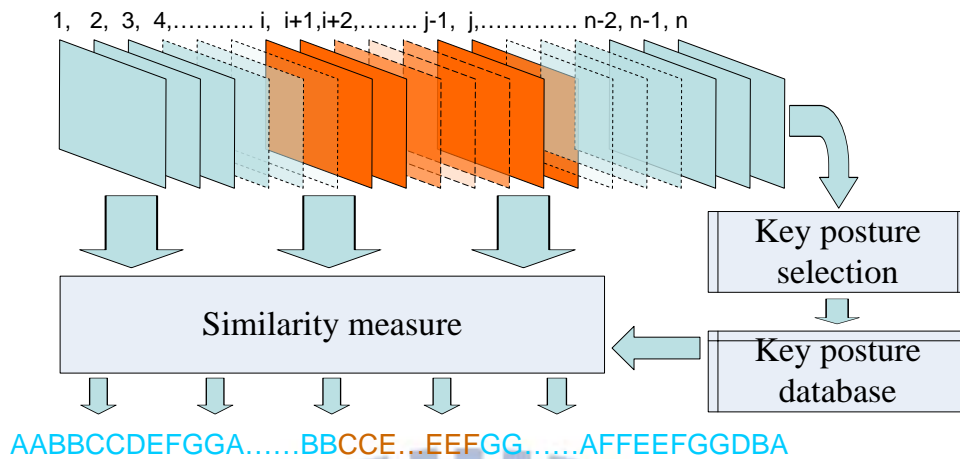


Figure 2.7 The process for converting available postures and virtual contours into a sequence of key posture labels. The blue frames and numbers indicate the frames with available postures and their corresponding key-posture labels. The orange frames and numbers indicate the frames with constructed virtual contours and their corresponding key-posture labels.

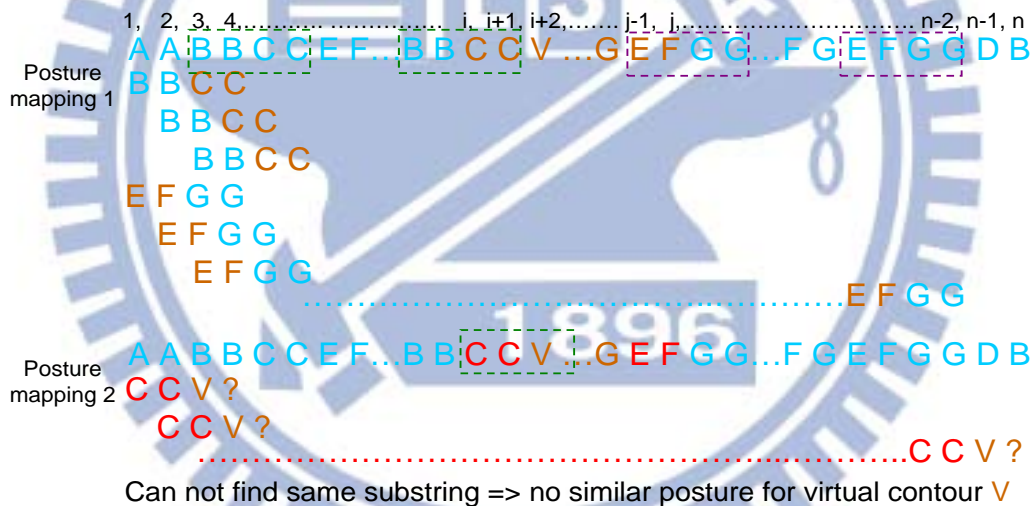


Figure 2.8 Examples of using substring matching to solve the posture mapping problem. The length of the substring is 4. The blue numbers indicate the key-posture labels of available postures; the brown numbers indicate the labels of virtual contours; and the red numbers indicate the labels of available postures used to replace the occluded objects. In the first posture mapping, the available postures in frames 5, 6, $n-5$ and $n-4$ are deemed the best matches to replace the damaged objects in frames i , $i+1$, $j-1$ and j respectively. In the second mapping, however, a good match cannot be found for the damaged object in frame $i+2$ (with the virtual contour labeled “V”).

2.2.5 Synthetic Posture Generation

The occlusion problem occurs in real-world applications all the time; hence, a virtual contour generated from an occlusion event may not find a good match among the selected key postures due to the lack of available non-occluded object postures. The problem of insufficient postures usually arises when the occlusion period for a to-be-completed object is long, resulting in many reconstructed virtual contours, or when the object's non-occlusion period is too short to collect a sufficiently rich set of non-occluded postures. Using a poorly matched posture to complete an occluded object can result in visually annoying artifacts. To resolve the problem where a virtual contour cannot find a good-match in the available key-posture database, we synthesize more postures by combining the constituent components of the available postures to enrich the content of the database. Figure 2.9 shows how a new posture is synthesized by using three constituent components (the head, torso, and legs) from different available postures selected by a skeleton matching process.

The flowchart of the proposed synthetic posture generation process is shown in Figure 2.10. First, the skeleton of a virtual contour that cannot find a good match in the posture database is extracted using the scheme

proposed in [28], which is also used to extract the skeletons of all available postures. Then, the constituent components of each selected key-posture are decomposed based on the distribution of the variance in alignment errors between every two aligned key-postures. The component decomposition result of key postures is used to help segment the extracted skeletons into their constituent components. We use the segmented skeleton components of a virtual contour to retrieve similar posture components, which are then used to synthesize new postures.

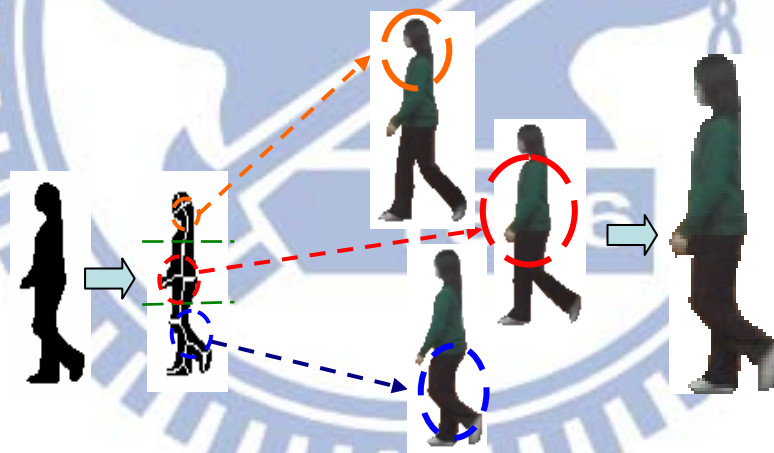


Figure 2.9 Synthesizing a new posture using available postures. The new posture is comprised of three components (the head, body, and legs) taken from different postures.

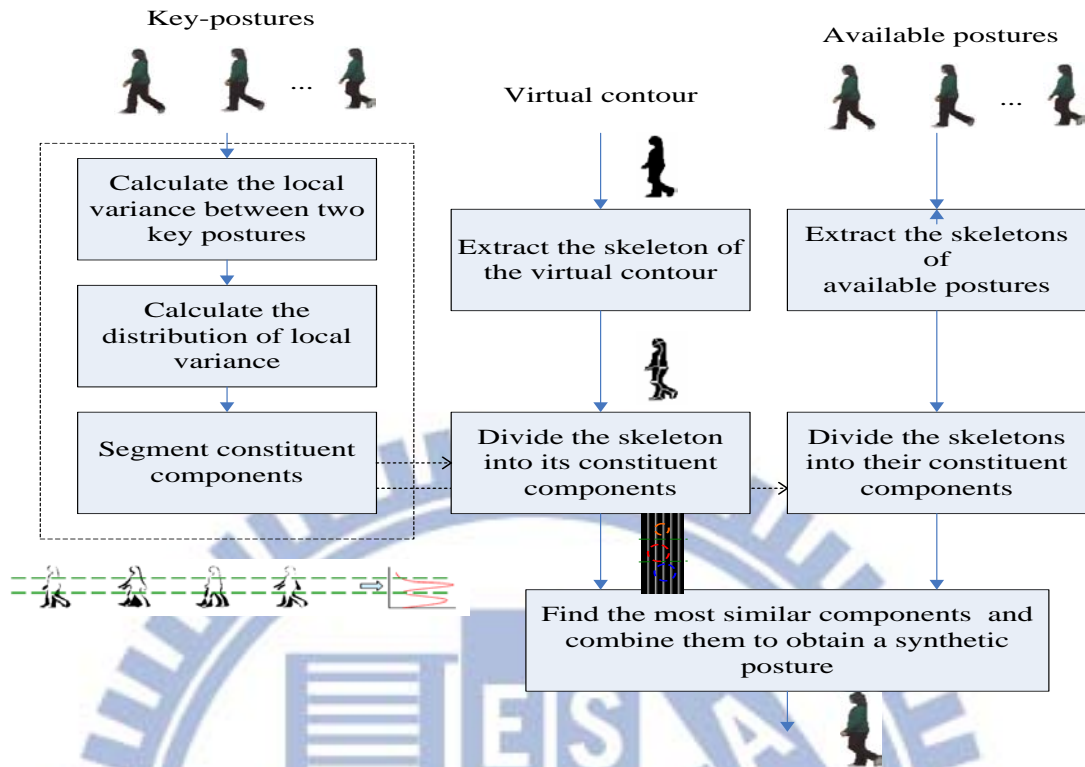


Figure 2.10 Flowchart of the proposed synthetic posture generation process.

All of the above-mentioned constituent components are derived from the components of existing database postures. To use these components, we need to perform segmentation on the key-postures in advance, as shown in Figure 2.11. After aligning the postures, we compute the difference between every two consecutive key postures. From the distribution of the variance, it is possible to identify the components that move more frequently. Then, we label the “frequently moving” components as the constituent components of the posture synthesis process.



Figure 2.11 The constituent components of a posture are partitioned based on local variance extraction. The dashed lines which separate postures into constituent components are determined based on the distribution of local variance shown on the right-hand side.

We use the skeletons of objects to retrieve similar posture components, which are then used to synthesize new postures. To extract object skeletons, we employ the method proposed in [28]. It defines candidate skeleton points as the centers of the maximal disks located inside the planar shape. Then, a Euclidean distance map is used to determine whether or not a candidate skeleton point is a genuine skeleton point. A candidate skeleton point is deemed a real skeleton point if any one of its eight neighbors satisfies the connectivity criterion:

$$\frac{r_2^2 - r_1^2}{\max(x, y)} < 1 \text{ and } D^2 \geq \rho, \quad (2.7)$$

where $x = |x_2 - x_1|$ and $y = |y_2 - y_1|$, in which (x_1, y_1) and (x_2, y_2) denote, respectively, the coordinates of the two nearest contour points e_1 and e_2 ; r_1 and r_2 represent, respectively, the shortest and longest distances between the contour point and the neighbors of the skeleton point; D is

the distance between the two nearest contour points; and ρ is a pre-determined threshold.

We use the following relevance metric, K , to measure the contribution of an arc to the shape of a contour in order to determine whether the arc is a redundant branch of the skeleton:

$$K(l_1, l_2) = \frac{\beta(l_1, l_2)l(l_1)l(l_2)}{l(l_1) + l(l_2)}, \quad (2.8)$$

where l_1 and l_2 represent, respectively, two line segments of the object's contour; $\beta(l_1, l_2)$ is the turn angle at the common vertex of segments l_{s_1} and l_{s_2} ; and $l(\cdot)$ denotes the length function.

The relevance metric allows us to select and remove arcs that only make a small contribution to an object's shape. This operation reduces the shape's contour, which is then used to remove unimportant skeleton points. We use the thresholds derived in the posture classification step to separate the skeletons of virtual contours and those of the available postures. After aligning the parts of a skeleton in the virtual contours with the corresponding parts in the available postures, the best-matched skeleton components of the available postures can be identified based on the following similarity metric:

$$S(T, S) = \sum_{(t_{x,y} \in T) \cap (s_{x,y} \in S)} w(t_{x,y}, s_{x,y}), \quad (2.9)$$

where T and S denote, respectively, the skeleton component of a virtual contour and the corresponding part in an available posture; and $w(t_{x,y}, s_{x,y})$ represents the matching score of the corresponding skeleton points, $t_{x,y}$ and $s_{x,y}$, of the virtual contour and the available posture, defined as follows:

$$w(t_{x,y}, s_{x,y}) = \begin{cases} score_1, & \text{if } t_{x,y} \text{ and } s_{x,y} \text{ belong to the skeleton region} \\ score_2, & \text{if } t_{x,y} \text{ and } s_{x,y} \text{ belong to the foreground region,} \\ 0, & \text{otherwise} \end{cases} \quad (2.10)$$

Here, the two score constants, $score_1$ and $score_2$, are set empirically as 3 and 1 respectively.

Finally, a new posture can be synthesized by combining all the best-matched constituent components of the available postures selected by the component-wise skeleton retrieval process.

2.3 Experimental Results

We used six test sequences to evaluate the efficacy of our method. Five sequences were captured by a commercial digital camcorder with a frame rate of 30 fps, and a resolution of 352×240 (SIF). The remaining one was taken from [1]. In the experiments, we first removed unwanted objects and occluded objects completely, and then used the proposed inpainting method to reconstruct the occluded objects.

Figure 2.12(a) shows some snapshots of test sequence #1, which contains a pedestrian. In this experiment, we intentionally removed the person from 20 consecutive frames, and then used the proposed method to restore the missing person. This test case simulates a real-world situation in which objects in a number of consecutive frames are damaged due to packet loss during transmission of the video (e.g., the loss of several video-object-planes of an MPEG-4 stream), or due to a damaged hardware component (e.g., a hard disk or an optical disk). Since we have the ground-truth of the missing object in this case, we can evaluate the performance of our object completion method based on the ground-truth. First, we observe that the virtual contours of the missing objects, constructed by combining the completed spatio-temporal slices (shown in Figure 2.12(b)), retain most of the objects' posture information. This verifies that the virtual contour of a missing object provides a fairly good initial estimate for finding the best-matched available posture to complete the missing object. Figure 2.12 (c) shows that the objects completed frame-by-frame by the proposed posture mapping scheme conform to the ground-truths very well. Moreover, the scheme maintains the temporal continuity of object motion even if the object is lost completely in several

consecutive frames.

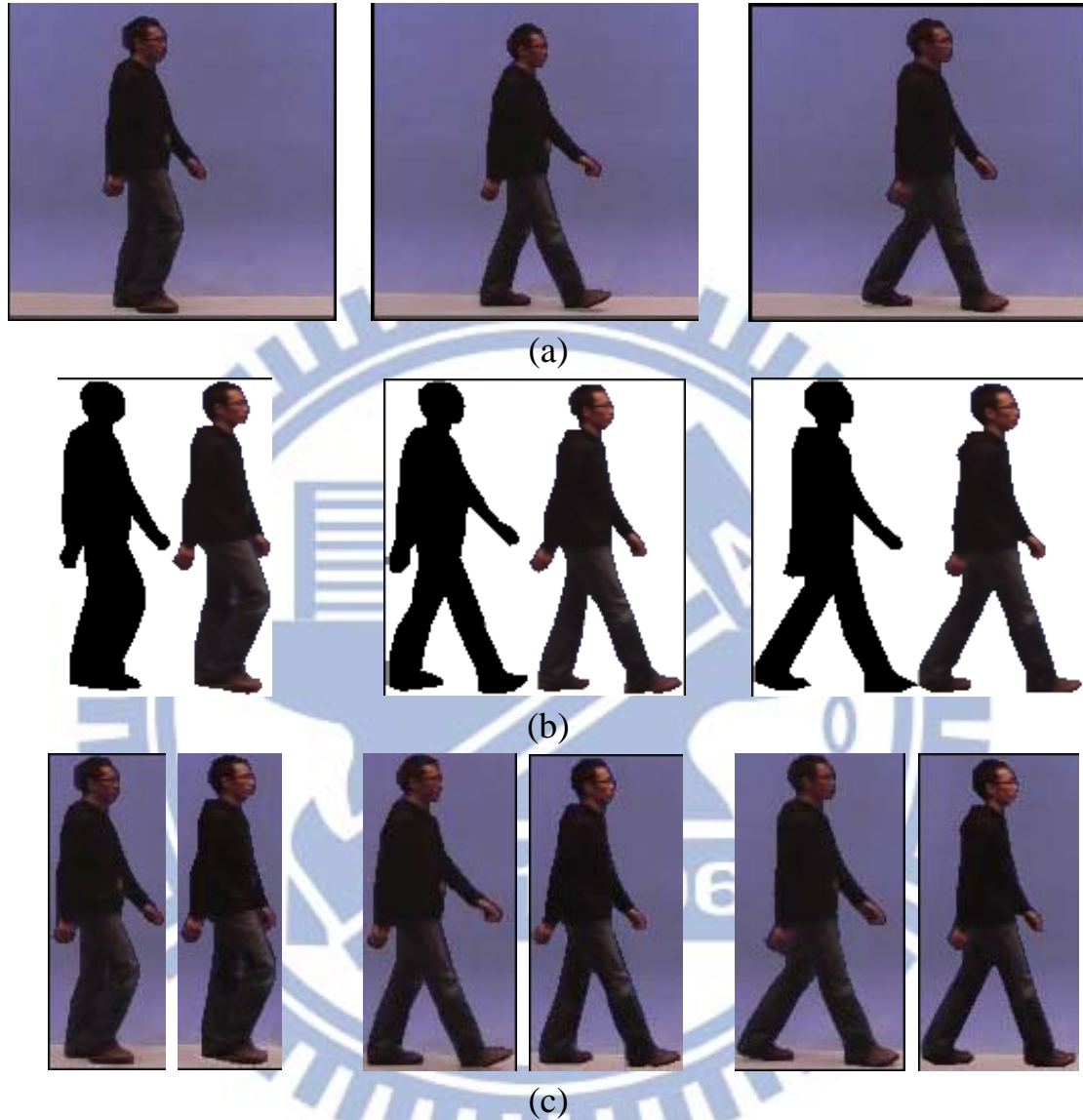


Figure 2.12 Test sequence #1 containing a single pedestrian: (a) some snapshots of the original video (ground-truths); (b) the virtual contours (on the left), which are constructed by combining the completed spatio-temporal slices and their corresponding best-match postures (on the right); (c) the corresponding completed frames; (d) comparison of the completed objects (on the left) and the ground-truths (on the right)

Test sequence #2, shown in Figure 2.13 (a), simulates a common real-life situation that occurs in home videos, i.e., two people walking

toward each other. In this scenario, one person is occluded by the other, which is not desirable. This case is similar to the situation where a moving object is occluded by a stationary object. After removing the unwanted object, we use the proposed method to restore the partially/completely occluded object. Figure 2.13 (b) shows, once again, that the virtual contours of damaged objects provide reasonably good estimates of the objects' postures. We do not have a ground-truth for this test sequence. However, Figure 2.13 (c) shows that the restored person moves with rather natural and continuous postures. Besides, our method maintains the temporal motion continuity of the object well. Note, the occluded girl turns her body a bit (i.e., the pose angle is changed) during the occlusion period. Since the pose angles of available postures are slightly different from the actual ones, the occluded objects are replaced with the available postures with similar silhouette information but different pose angles, leading to some artifact during the transition of pose angle (see the video in [30]). Such pose angle change problem has not yet been addressed in this work.

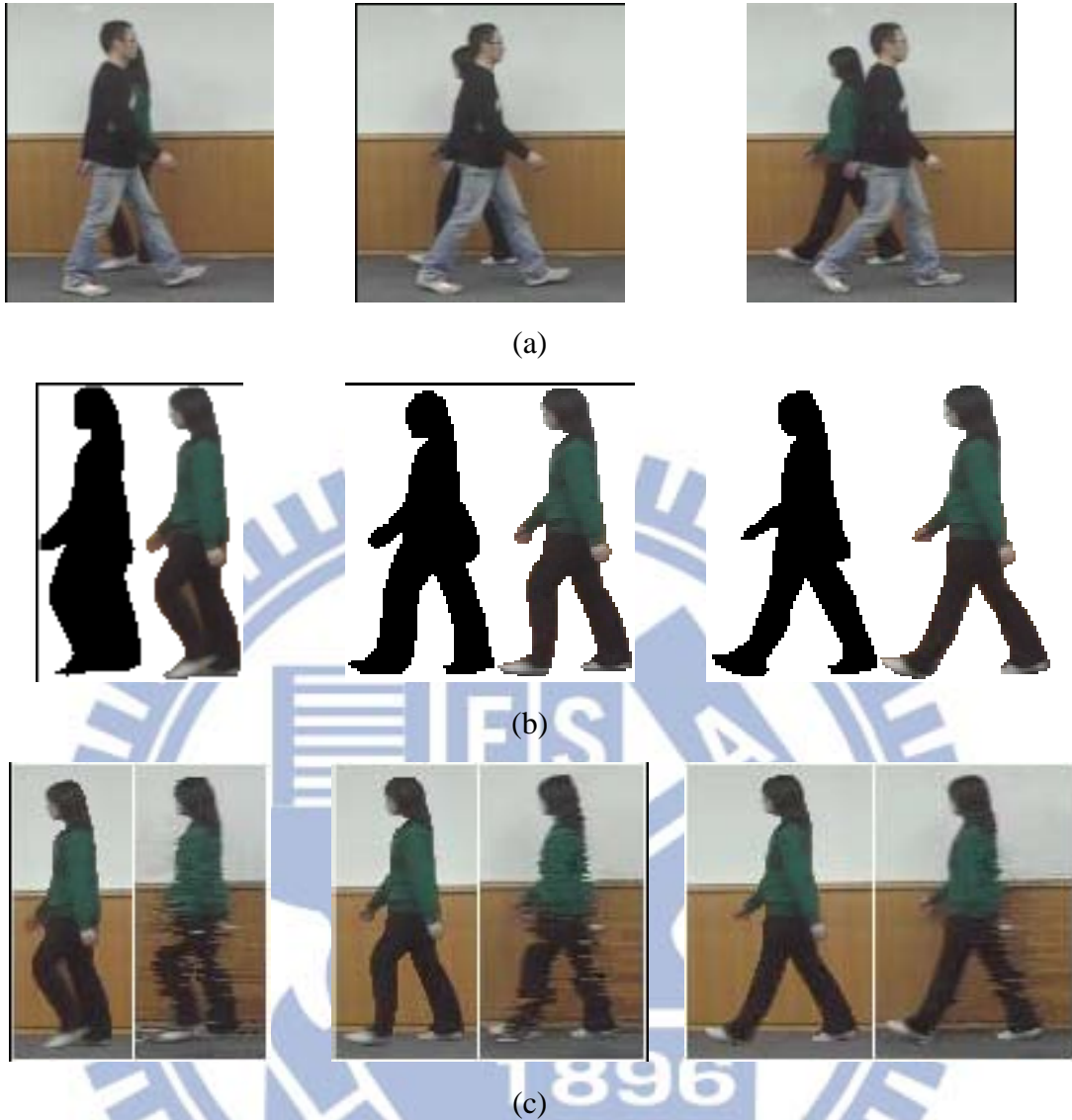


Figure 2.13 Test sequence #2 with two people walking toward each other: (a) original video frames; (b) the virtual contours (on the left), which are constructed by combining the completed spatio-temporal slices and the corresponding best-match postures (on the right); (c) the completed frames (on the left) using the original key-postures and the additional synthetic postures and the corresponding frames composed from the completed 2-D slices (on the right).

Test sequence #3 (Figure 2.14 (a)) is similar to test sequence #2, except that the person is occluded for a significantly longer period than in sequence #2. The longer occlusion period made it difficult to complete the occluded object because only a small number of available

non-occluded postures were available in the sequence. In other words, the key-postures selected from the available postures were not sufficiently comprehensive, so we could not find a good match among the key-postures for the occluded object. Figure 2.14 (b) shows the virtual contours of the occluded object and its corresponding matched postures. The postures matched with the set of insufficient available postures appear to be incorrect in the hands and legs, leading to visually unpleasant artifacts in the completed video. Recall that our scheme minimizes the effect of insufficient available postures by adding synthetic postures to the available posture database to enrich the choice of postures, as shown in Figure 2.14 (c) and Figure 2.14 (d).



(a)

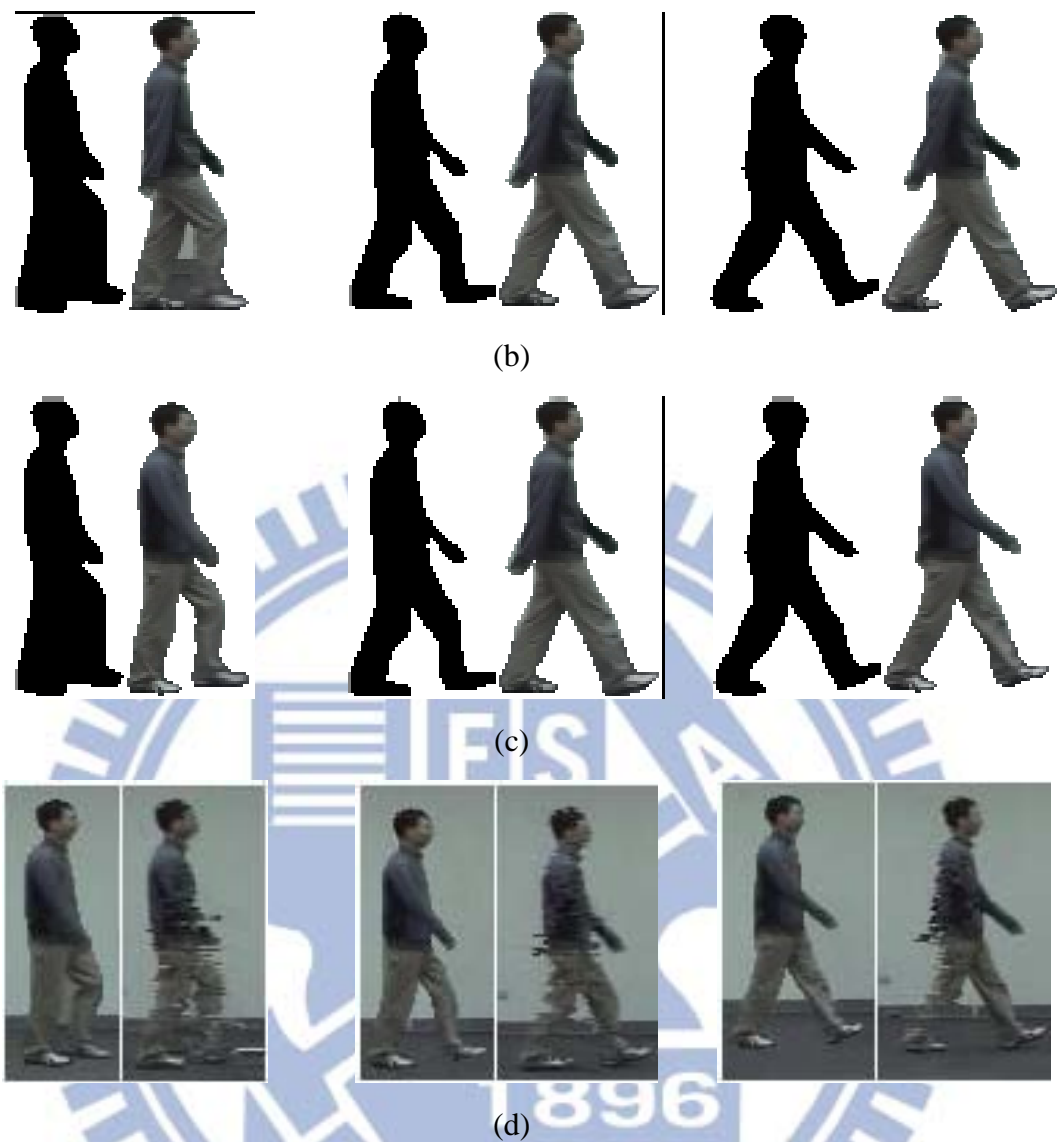


Figure 2.14 Test sequence #3 containing two people walking toward each other (with a long occlusion period): (a) original video frames; (b) the virtual contours (on the left) and the corresponding best-match postures (on the right) without including synthetic postures; (c) the virtual contours (on the left) and the corresponding best-match postures (on the right) with the additional synthetic key-postures; and (d) the completed frames (on the left) using the original key-postures and the additional synthetic postures and the corresponding frames composed from the completed 2-D slices (on the right).

Test sequence #4 shown in Figure 2.15 (a), also shows two people walking toward each other, where the subject moves both horizontally and vertically. Moreover, the subject changes direction leading to

non-linear motion and change of object size. In this scenario, we perform posture alignment/normalization prior to sampling the 2-D spatio-temporal slices. After removing the unwanted object, we use the proposed method to restore the occluded object. Figure 2.15 (c) shows that, even with non-pure horizontal motion and non-linear motion, the proposed method is still effective in maintaining the spatial consistency and temporal continuity.

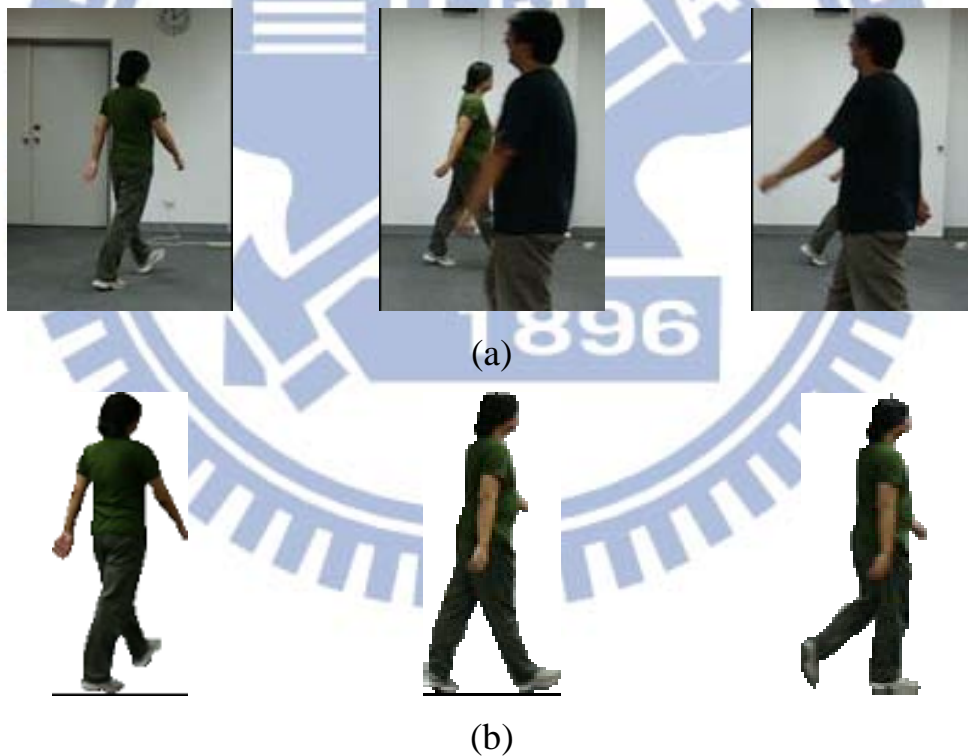




Figure 2.15 Test sequence #4: (a) some snapshots of the original video; (b) the corresponding best-match postures; and (c) the result derived by the proposed method.

The proposed system was implemented on a PC equipped with Intel Core2 Duo CPU 2.83GHz and 3.5 GB system memory. The codes (implemented in MATLAB) for patch -based image inpainting and skeleton generation are obtained from [16] and [28], respectively. The remaining codes are all implemented in C++. The run time of each step for each test sequence is listed in TABLE I. In the four test sequences, the number of available postures in sequence #3 is not rich enough to achieve satisfactory object inpainting performance. Therefore the synthetic posture generation process is used to improve the performance.

Table 2.1 Run-time analysis of key operations in the proposed method

	Virtual contour generation	Posture mapping	Synthetic posture generation
Sequence #1	838.53 s	9.82 s	not used
Sequence #2	181.56 s	9.36 s	not used
Sequence #3	195.64 s	9.06 s	82.89 s

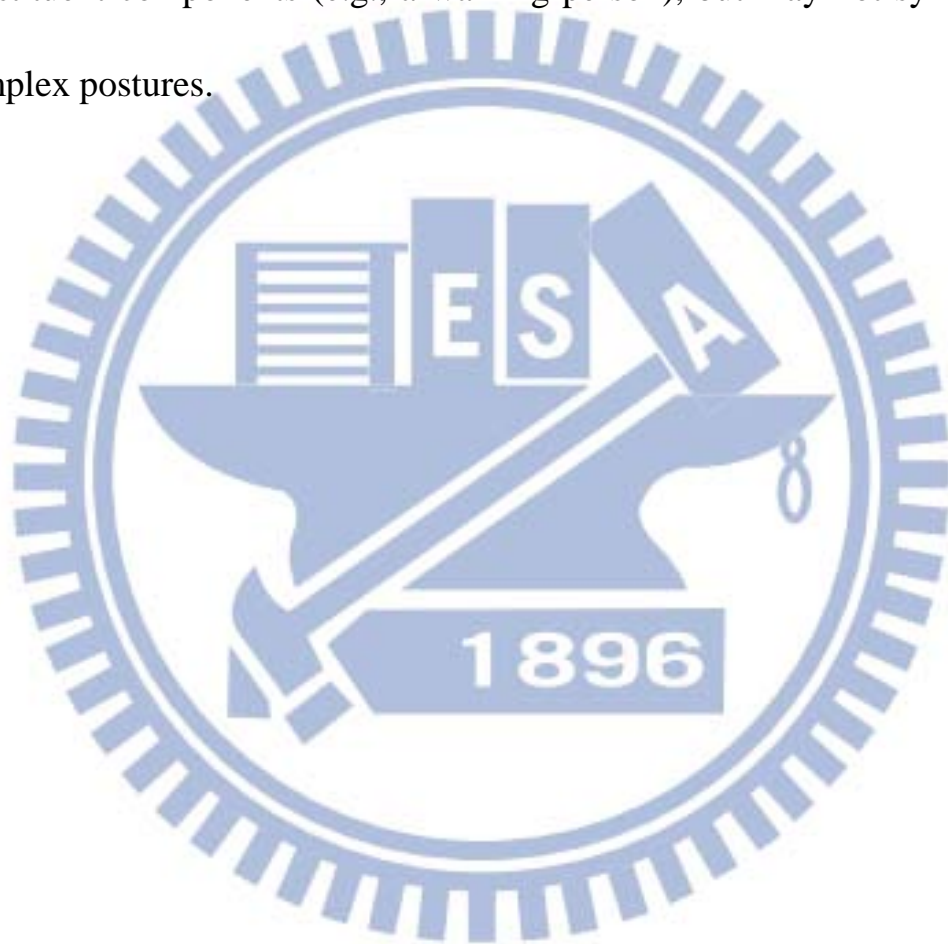
Sequence #4	608.16 s	9.21 s	not used
-------------	----------	--------	----------

2.4 Summary

To resolve a number of problems related to video completion, in this Chapter we propose a novel method that handles the completion of objects and completion of the background separately. The method is comprised of three steps: virtual contour construction, key posture-based sequence retrieval, and synthetic posture generation. An efficient posture mapping method has been proposed that uses key posture selection, indexing, and coding operations to convert the posture sequence retrieval problem into a substring matching problem. In addition, we have developed a synthetic posture generation scheme that enhances the variety of postures available in the database. Our experiment results show that the proposed method generates completed objects with good subjective quality in terms of the objects' spatial consistency and temporal motion continuity.

The proposed method still has a few constraints. First, if an object moves nonlinearly during an occlusion period, the virtual contour construction may not compose sufficiently accurate postures. But should there be enough non-occluded portion of the object, the linear motion

constraint may be relaxed. Second, currently the proposed method does not deal with the illumination change problem that occurs if lighting is not uniform across the scene. Third, the synthetic posture generation method can only deal with objects that can be explicitly decomposed into constituent components (e.g., a walking person), but may not synthesize complex postures.



Chapter 3

Human Object Inpainting Using Manifold Learning-Based Posture Sequence Estimation

In this Chapter, we propose a framework for virtual contour guided video object inpainting using posture mapping and retrieval. First, we give an introduction about this research topic. The proposed approach is then described. Next, we detail the experiment results. Finally, we present our conclusions.

3.1 Introduction

Video inpainting [1]-[11] is a popular research field in recent years owing to its powerful capability in video editing and recovering. A number of algorithms for automatic video inpainting have been proposed in the past few years. Conventional video inpainting methods can be roughly classified into two types: the first type is patch-based [1]-[6] and the other type is template-based [7][8]. However, patch-based approaches would cause spatial or temporal structure inconsistency artifacts and template-based approaches would cause temporal discontinuity.

Recently, Ding *et al.* [10] proposed a nonlinear dimension

reduction-based video inpainting technique that utilizes Local Linear Embedding (LLE) [31] to transform data observed in frames into embedded features in a low-dimension manifold. Then, the embedded features are organized to form a Hankel matrix and missing data can be determined by minimizing the rank of the matrix. Finally, the Radial Basis Function (RBF) is used for inverse mapping. Again, the drawback of this method is that it may cause blurring and ghost image artifacts if the object's motion is not periodic.

Motion prior models derived from training data have also been successfully applied in applications of marker-free human motion capture and analysis [31]–[34]. Generally two main classes of motion priors can be identified [34]. The first class utilizes an explicit motion model to guide motion analysis and tracking of body parts. For example, the method proposed in [35] utilizes Variable length Markov Models (VLMM) to characterize both the short-term dynamics and long-term history of video data. Similar to Ding *et al.*'s approach [10] and this work, the second class learns a low-dimensional posture manifold and performs analysis and tracking in the low-dimensional manifold [36][37]. The inverse mapping from the low-dimensional manifold to the

high-dimensional full body configuration can be accomplished via RBF or Locally Linear Coordination (LLC) [38]. Although the basic components for dimensionality reduction and inverse mapping are similar, as motion analysis is aimed at tracking of human motion, the key component of object inpainting—recovering missing trajectories in the learned low-dimensional manifold, was usually not addressed in these motion analysis works.

Our literature survey shows that most video inpainting algorithms generate artifacts if the object to be inpainted is completely occluded or its motion is not periodic. To void generating such artifacts, a posture sequence estimation process of good accuracy is required for object inpainting. To this end, Xu *et al.* [39] proposed a method for animating animal motions. The model rearranges available animal templates to form a new animal motion sequence by minimizing a predefined energy function. In this work, rather than using an optimization approach, which is time consuming, we propose a posture sequence estimation method that maintains the continuity of local motion. The proposed framework consists of three steps: human posture synthesis, graphical model construction, and posture sequence estimation. Human posture synthesis

is used to enrich the number of postures in the database, after which all the postures are used to build a graphical model that can predict motion tendency. We also propose two constraints to confine the motion continuity property. The first constraint limits the maximum search distance if a trajectory in a graphical model is discontinuous; and the second confines the search direction in order to maintain the tendency of an object's motion. We perform both forward and backward prediction to derive local optimal solutions. Finally, we apply the Markov Random Field model to compute an overall best solution, and the potential trajectory with the maximum total probability is taken as the final result. The proposed posture sequence estimation model can help identify a set of suitable postures from the posture database to restore damaged/missing postures. It can also make a reconstructed motion look continuous. The advantage of this posture sequence estimation strategy is that it can handle cases like non-periodic motion or complete occlusion. These capabilities are powerful because conventional model-based motion prediction methods [31][40][41] must use a training process to achieve the same goal.

3.2 Human Object Inpainting Using Posture Sequence Estimation

In this section, we explain how to perform human object inpainting based on the proposed posture sequence estimation method. As mentioned earlier, the method includes three steps: posture synthesis, graphical model construction, and posture sequence estimation. We discuss the steps in detail in the following sections.

3.2.1 Human Posture Synthesis

The problem of an insufficient number of postures will affect the visual quality of any video sequence generated by a posture prediction-based approach. To solve the shortage-of-posture problem, we utilize our previous posture synthesis method [30] that was mainly designed for generating synthetic human postures to increase the number of postures.

3.2.2 Graphical Model Construction

After creating synthetic postures, the posture database will contain a lot of postures that can be used to build a graphical model of an object's motion, as shown in Figure 3.1. The model provides a simple

representation of an object’s motion. To obtain such a model, all postures (both synthesized and existing postures) must be projected onto a feature space. Then, we link the postures that appear in adjacent frames in the constructed feature space. After applying the above procedure, we can obtain a graphical representation of the object’s motion. To model the distribution of the postures in the feature space, we need to know the distances between distinct postures. We use a shape context descriptor that we developed in a previous work [24], that is a modified version of the descriptor proposed in [23], to compile a detailed description of each posture. The value of the shape context is calculated along the silhouette of the posture. In the posture sequence estimation stage, the values of the shape contexts will be used to compare the degree of similarity between two distinct postures.

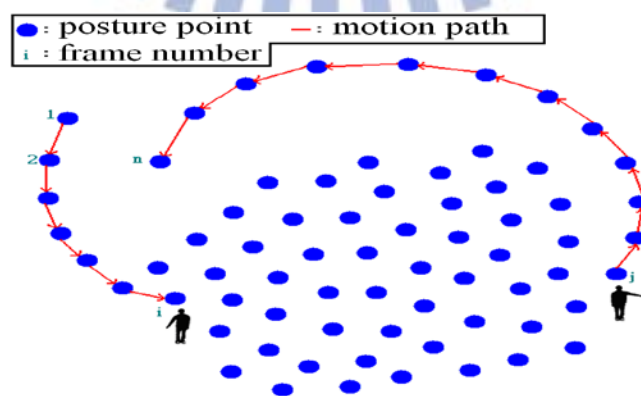


Figure 3.1 A graphical model of an object’s motion in a low-dimension manifold.

The blue points represent the feature points of the postures, and the red lines connect two feature points whose corresponding postures appear in adjacent frames. In this example, occlusion occurs between frames i and j , so we try to find a motion path with l internal points that can be used to link points x_i and x_j .

By using the context descriptor proposed in [23][24], we can calculate the degree of similarity between two distinct postures. Then, based on the similarity scores of the postures, we cluster all the postures in the database by using a nonlinear dimension reduction method called isometric feature mapping (Isomap) [44]. In our application, existing and synthesized postures are regarded as input data points for Isomap, and the distance between two data points is equivalent to the degree of similarity between two corresponding postures. We modify the Isomap algorithm to fit our requirements as follows:

Step 1) Construct a neighborhood graph: If x_i is one of the K -nearest neighbors (K -NN) of x_j , define a graph G that connects data points x_i and x_j . The length of the edge between x_i and x_j is used to measure the degree of similarity between postures o_i and o_j .

Step 2) Compute the shortest paths: Find the shortest path between each pair of feature points in G . The matrix $D_G = (d_G(x_i, x_j))$ contains all the shortest paths between all pairs of data points in G .

Step 3) Construct a d -dimensional embedding: Find the eigenvector

λ of the matrix $\Gamma(D_G)$ (The operator Γ is defined as $\Gamma(D_G) = a_{ij} + a_{**} - a_{*j} - a_{i*}$, where $a_{ij} = -\frac{1}{2}(d_G(x_i, x_j))^2$, $a_{i*} = \frac{1}{n} \sum_j a_{ij}$, $a_{*j} = \frac{1}{n} \sum_i a_{ij}$, and $a_{**} = \frac{1}{n^2} \sum \sum a_{ij}$). Then, to derive the final result, we apply classical Multi-Dimensional Scaling (MDS) [45] to the matrix of graph distances D_G .

3.2.3 Posture Sequence Estimation

Based on the graphical model of an object's motion shown in Figure 3.1, we obtain suitable postures to replace damaged/missing postures by finding an approximate path that links data points x_i and x_j in a low dimension manifold. Intuitively, a motion path can be reconstructed by taking the shortest path between two nodes or by an optimization process [39], but these two approaches cannot guarantee the smoothness of a recovered motion. To resolve the problem, we propose using two constraints to regulate the motion continuity property in the local region of a graphical model. Specifically, we need a strategy to select a certain number of data points that satisfy the continuous motion constraint. The first constraint limits the search range to within a reasonable neighborhood, as shown in Figure 3.2. Therefore, we need to define the

search range of the complete trajectory of an object's motion. In the manifold domain, such trajectories are comprised of a number of linked data points (see Figure 3.1). To determine the distance between any two consecutive data points on a trajectory, we calculate the shape context difference between their corresponding postures. Then, the maximum distance among all the measured distances is taken as the search range to satisfy the first constraint. Since the search range is circular, we calculate the radius as follows:

$$r = \max_{\forall e_{ij} \text{ on a complete trajectory}} e_{ij}, \quad (3.1)$$

where e_{ij} represents the distance between x_i and x_j on an object's motion trajectory.

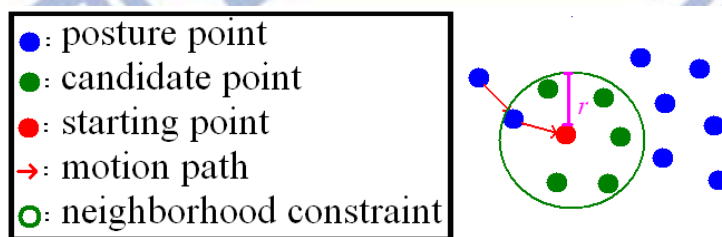


Figure 3.2 The neighborhood constraint.

The second constraint is introduced to maintain the tendency of an object's motion in each local region. It can be realized by checking the tendency of an object's motion trajectory in a graphical model. In a low

dimensional manifold, a motion trajectory does not change direction significantly in a neighborhood region. Based on this observation, a variance constraint of motion tendency is designed to ensure that the variance of motion tendency stays within a reasonable range (see Figure 3.3). In the manifold domain, the complete trajectory of an object's motion is comprised of a number of linked segments, as shown by the red lines in Figure 3.1. For the segments indicated by the lines, we compute the change in direction between any two consecutive segments based on the inner product of their corresponding vectors. Among all the computed direction changes, the largest direction change is taken as the maximum allowable angle for direction change. This angle, which is the basis for executing the second constraint, is calculated as follows:

$$\alpha = \max_{\forall \theta_{ijk} \text{ on a complete trajectory}} \theta_{ijk}, \quad (3.2)$$

where θ_{ijk} represents the angle between the vectors $\overrightarrow{x_i x_j}$ and $\overrightarrow{x_j x_k}$ on an object's motion trajectory.

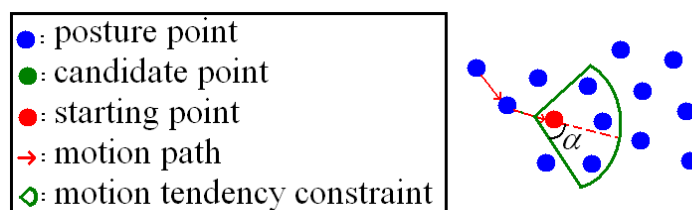


Figure 3.3 The motion tendency constraint.

The above constraints are designed to maintain the local continuity of an object's motion. To maintain the global motion continuity, we propose a two-way (forward-backward) prediction mechanism. We use three time instants, $t-1$, t , and $t+1$, to explain how the proposed mechanism operates. In the forward operation, we make a forward prediction on each data point at time $t-1$. The motion tendency constraint and the search range constraint are applied to determine m probable data points at the next time instant t . The selected data points, m , will be used to predict the candidate data points at time $t+1$. We apply the same strategy in the reverse direction and collect related information from $t+1$ to t , and from t to $t-1$. Then, we combine the results from the bi-directional processing to obtain the final results for time t . To illustrate the two-way prediction process further, we use a test sequence containing 245 frames. Some snapshots extracted from the test sequence, #1, are shown in Figure 3.4. The candidate points chosen at time instant 19 ($t-1$) are indicated by the blue dots in Figure 3.5(a), and their corresponding postures are shown on the left-hand side of the figure. Those candidate points are used to perform forward prediction. The predicted candidate points at time instant 20 are shown in Figure 3.5(b). We apply the same procedure in the reverse

direction and generate results from $t = 21$ to $t = 20$ (shown in Figure 3.5(c) and (d)). The two sets of results are then combined to form the final results, as shown in Figure 3.5(e). Table I provides detailed information about the above mentioned processes, including the distance and angle information calculated during the forward and backward prediction steps.

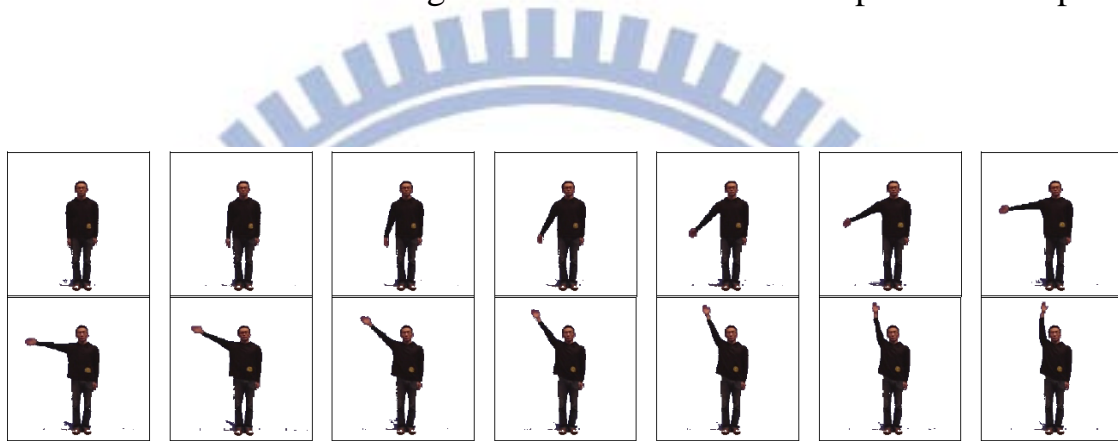
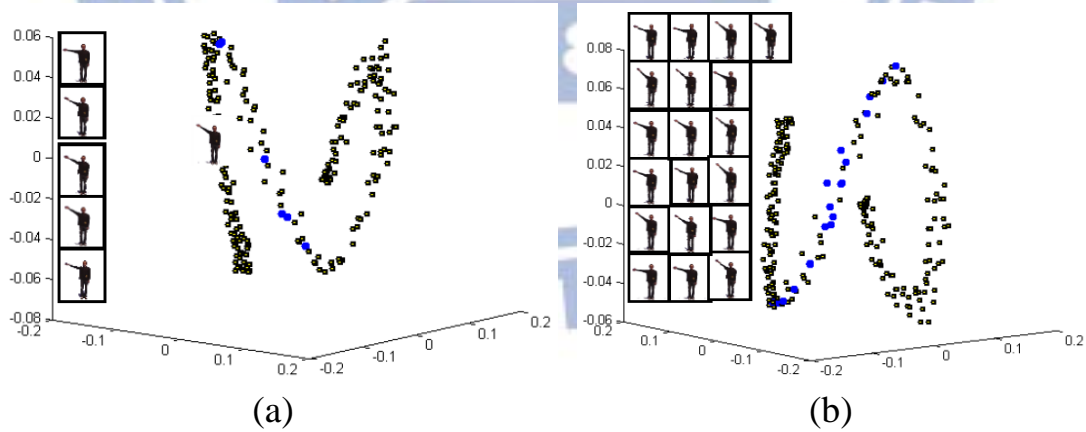


Figure 3.4 Some snapshots extracted from test sequence #1.



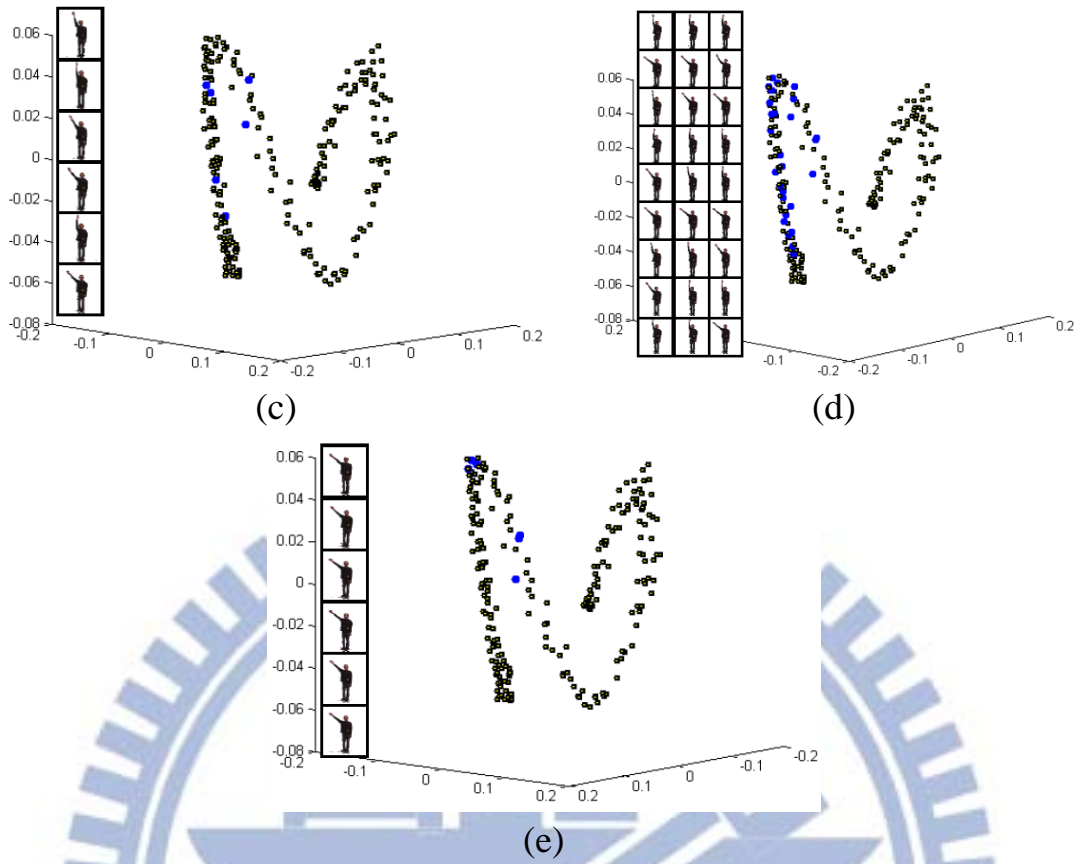


Figure 3.5 (a)–(b) some forward prediction steps, (c)–(d) some backward prediction steps, and (e) the combined results of two-way prediction at time t .

Table 3.1. Detailed information derived during the forward-backward prediction process

Forward prediction from time instant 19 to time instant 20 (D: distance; A: angle)													
T:19													
T:20													
	D:0.026	D:0.046	D:0.033	D:0.046	D:0.049	D:0.029	D:0.044	D:0.042	D:0.040	D:0.049	D:0.043	D:0.032	D:0.028
	A:13.92	A:31.96	A:40.45	A:14.75	A:27.74	A:5.957	A:8.784	A:19.02	A:28.32	A:19.01	A:44.11	A:31.16	A:15.99
T:19													

T:20	
	D:0.041 D:0.038 D:0.041 D:0.024 D:0.040 D:0.045 D:0.034 D:0.032 D:0.033 D:0.048 D:0.043 D:0.049 D:0.035
	A:37.53 A:8.025 A:8.587 A:8.547 A:6.434 A:24.20 A:4.064 A:38.05 A:35.21 A:22.38 A:12.17 A:22.67 A:24.53
Backward prediction from time instant 21 to time instant 20 (D: distance; A: angle)	
T:21	
T:20	
	D:0.031 D:0.041 D:0.039 D:0.049 D:0.031 D:0.041 D:0.043 D:0.046 D:0.043 D:0.028 D:0.045 D:0.048 D:0.037
	A:26.12 A:2.623 A:19.61 A:4.843 A:15.55 A:26.61 A:29.67 A:18.70 A:13.51 A:13.04 A:4.683 A:5.593 A:13.18
T:21	
T:20	
	D:0.045 D:0.041 D:0.047 D:0.031 D:0.043 D:0.023 D:0.049 D:0.047 D:0.049 D:0.032 D:0.048 D:0.041 D:0.035
	A:42.77 A:43.72 A:31.33 A:49.49 A:48.23 A:1.048 A:33.80 A:23.55 A:9.623 A:14.78 A:5.354 A:5.704 A:0.914

Since the motion continuity constraint is only effective on local regions, we use the Markov Random Field (MRF) model to derive global motion continuity. MRF provides a convenient and accurate way to model context-dependent entities, such as image pixels and correlated features. The above modeling can be achieved by characterizing the mutual influences that relate such entities. To predict an object's motion, instead of following the Markov assumption, we assign one node of the Markov network to each time state. Then, the constructed network can reflect

statistical dependencies. Given a set of data points located at the intervening nodes, every node of a Markov network is statistically independent of other nodes in the network. Since our Markov network does not contain loops, the above-mentioned Markov assumption results in simple “message-passing” rules for computing the probability during inference. The data point estimated at node j is

$$c_j^* = \arg \max_{c_j} p(c_j) M_j^{j-1} M_j^{j+1}, \quad (3.3)$$

where c_j denotes the candidate point associated with node j , $p(c_j)$ is the self probability of candidate point c_j , and M_j^{j+1} is the message derived from node $j-1$ to node j . M_j^{j+1} can be calculated as follows:

$$M_j^{j+1} = \max_{[c_k]} \Psi(c_j, c_{j+1}, c_{j+2}) p(c_{j+1}) \tilde{M}_{j+1}^j \tilde{M}_{j+1}^{j+2}, \quad (3.4)$$

where \tilde{M}_{j+1}^j is the previous message, which is used to generate M_j^{j+1} by executing (3.4). M_j^{j+1} includes the probability information of all the candidate data points of node k . The initial \tilde{M}_{j+1}^j message is set as a column vector with the initial probability of all the elements associated with node j . The function $\Psi(c_j, c_{j+1}, c_{j+2})$ is defined as follows:

$$\Psi(c_j, c_{j+1}, c_{j+2}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\theta - \mu)^2}{2\sigma^2}\right), \quad (3.5)$$

where θ is the angle between vectors $\overrightarrow{c_j c_{j+1}}$ and $\overrightarrow{c_{j+1} c_{j+2}}$; and μ and σ are the mean and standard deviation of all angles in a complete trajectory of

an object's motion.

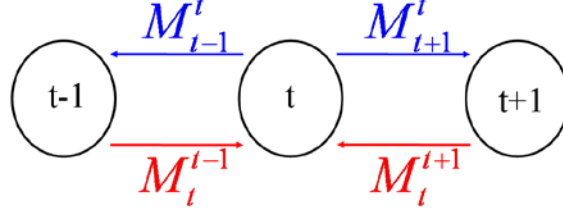


Figure 3.6 An example of the MRF process.

To better explain how (3.3), (3.4) and (3.5) find an optimal c_t^* , we use the three nodes shown in Figure 3.6 as an example.

Initially, node t receives two messages in the form of a column vector with the initial probabilities of the elements associated with node $t-1$ and $t+1$. It then sends the two messages, M_{t-1}^t and M_{t+1}^t , to nodes $t-1$ and $t+1$ respectively. The messages contain the probability information of all the candidate data points associated with node t . Before the information is sent, it is reordered to form a column vector. On receipt of the information, nodes $t-1$ and $t+1$ respond by sending messages M_t^{t-1} and M_t^{t+1} , respectively, to node t . When each candidate point of node t receives the message M_t^{t-1} it finds a matching point in node $t-1$ as follows:

$$\hat{p}(c_t) = \arg \max_{c_{t+1}} \Psi(c_{t-1}, c_t, c_{t+1}) p(c_{t-1}) p(c_t) p(c_{t+1}), \quad (3.6)$$

where $\hat{p}(c_t)$ is the new self probability of candidate point c_t , $p(c_t)$ denotes

the previous self probability of candidate point c_t , and $p(c_{t-1})$ and $p(c_{t+1})$ are the probabilities propagated by messages M_t^{t-1} and M_t^{t+1} , respectively. After normalizing the probability value of each candidate point calculated by (3.6), we obtain a new probability value for each candidate point. Then, node t sends the updated message M_{t+1}^t with the new probability to node $t+1$. Similarly, if node t receives an updated message from node $t+1$, the probability values of all the candidate points of node t are recomputed and sent to node $t-1$. Freeman *et al.* [53] showed that after, at most, one global iteration of (3.4) on each node of the network, (3.3) can derive the desired optimal estimate of c_j^* at node j .

3.3 Experimental Results

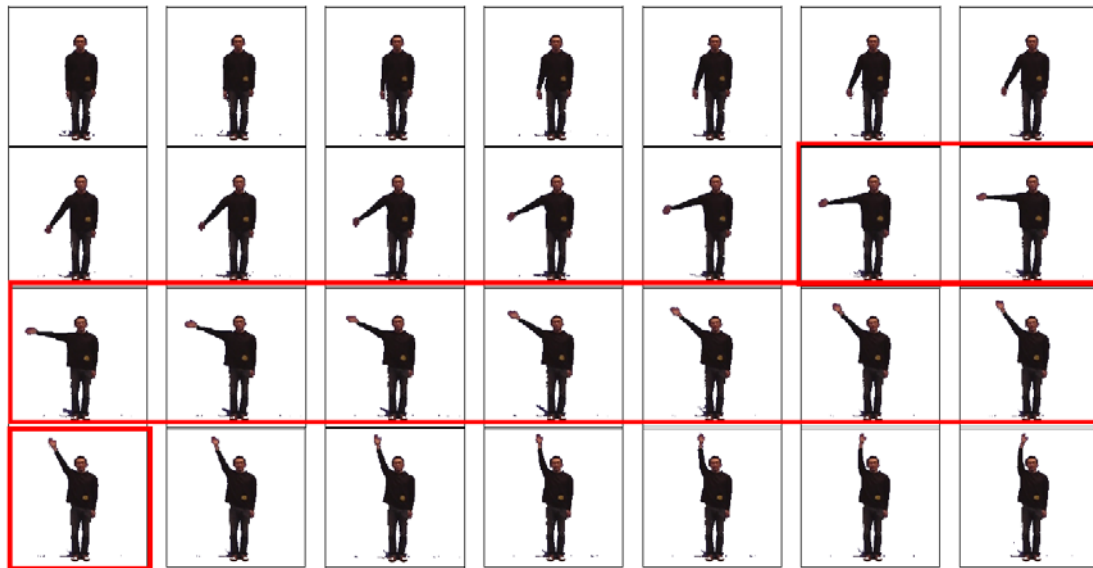
To test the effectiveness of the proposed posture sequence estimation method, we performed experiments on eight test sequences, where part of them were captured with a camcorder and the remaining were grabbed from the Weizmann database [46] and the Internet. In addition to test sequence #1 shown in Figure 3.4, we used sequences #2 and #3 to evaluate the proposed method. In the experiments, we first removed several consecutive frames to simulate a real-world situation where

objects in a number of consecutive frames were damaged due to packet loss. Then, we applied the proposed posture sequence estimation method to reconstruct the motion of each object. We also compared the performance of our approach with that of Ding *et al.*'s approach [10] and Xu *et al.*'s approach [39]. For all the test sequences, the proposed method maintained the motion continuity of a reconstructed motion and yielded better results than the compared approaches.

In the first experiment, we removed 10 of the 245 frames in test sequence #1. Part of the sequence (28 frames) is shown in Figure 3.7(a). In the Figure 3.7(a), the 10 frames that we removed are bounded by the red rectangle. Figure 3.7(b), (c), and (d) show the missing sequence that was reconstructed by applying Ding *et al.*'s approach [10], Xu *et al.*'s approach [39] and our approach, respectively; and Figure 3.7(e) shows the corresponding trajectories reconstructed by the three approaches in the manifold space. Among the trajectories, the red, blue, yellow and green colors represent the ground-truth trajectory, and the trajectories reconstructed by Ding *et al.*'s approach, Xu *et al.*'s approach and the proposed approach, respectively. We observe that the trajectory reconstructed by our approach maintains the best motion continuity; and

it is also the smoothest of the three trajectories. Because the proposed posture sequence estimation method is more effective in recovering an object's motion and maintaining motion continuity simultaneously, we conclude that it is more suitable for object inpainting than the compared methods.

Table 3.2 details the results of the ground-truth and the three compared methods. The top row shows the sequence of missing ground truth postures; and the second, third, and fourth rows show the missing frames reconstructed by Ding *et al.*'s method, Xu *et al.*'s method, and our method, respectively. The black parts of the figures are the ground-truth postures; the gray parts are perfectly matched portions; and the red parts belong to reconstructed postures. We observe that the frames reconstructed by our method are consistently better than those derived by the compared methods.



(a)



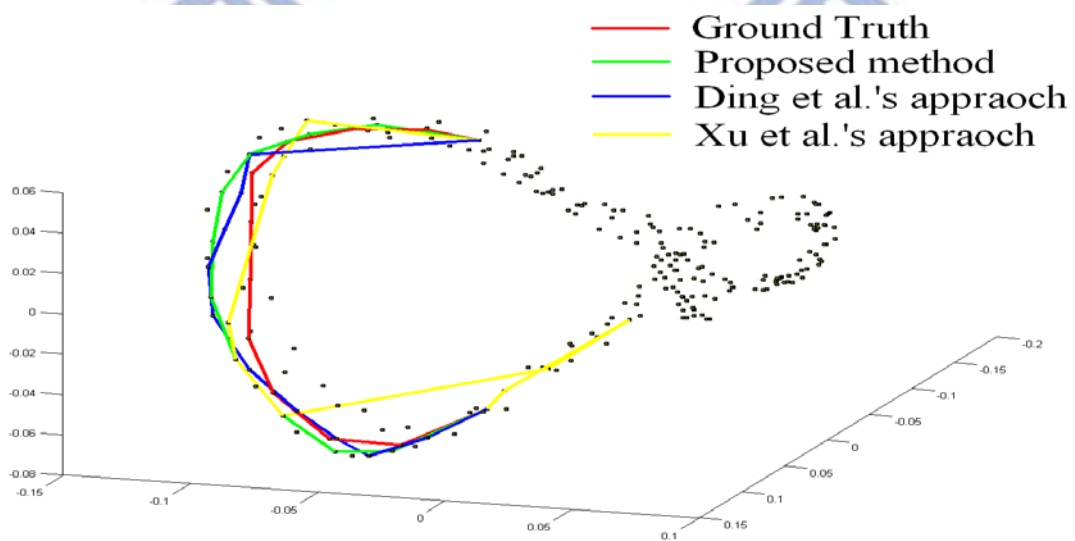
(b)



(c)















































(d)



(e)

Figure 3.7 The experiments on test sequence #1: (a) partial sequence of test sequence #1 in which the red rectangle indicates missing frames; (b) frames reconstructed by Ding *et al.*'s approach; (c) frames reconstructed by Xu *et al.*'s approach; (d) frames reconstructed by the proposed approach; and (e) the corresponding trajectory information of predicted object motion generated by the three approaches.

Table 3.2 Comparison of the ground-truth postures and the reconstructed missing postures (The parts in black, red and gray represent the ground-truth postures, reconstructed postures, and perfectly matched portions, respectively)

Ground-truth												Average
Ding <i>et al.</i> [10]												91.4%
	94.7%	93.0%	91.5%	90.7%	91.1%	90.8%	90.6%	90.6%	90.8%	90.6%		
Xu <i>et al.</i> [39]												89.1%
	89.8%	87.8%	85.6%	85.8%	89.5%	90.4%	88.0%	93.1%	90.6%	91.2%		
Ours												96.3%
	98.5%	97.7%	96.7%	96.8%	96.5%	95.5%	96.3%	96.4%	92.7%	96.4%		

In the second experiment, we used test sequence #2, which contained 100 frames. In the sequence, two people are walking toward each other, and one person occludes the other in about 20 frames (some of the frames are shown in Figure 3.8(a)). Figure 3.8 (b), (c), and (d) show, respectively, the snapshots of human objects reconstructed by the methods in [30] and [39] and our approach. From the reconstructed frames, it is apparent that our approach was the most effective in recovering the occluded frames.

Using the recovered sequence generated by our approach yielded the best inpainting results among the three compared approaches, as shown in Figure 3.8(e).



(a)



(b)



(c)



(d)



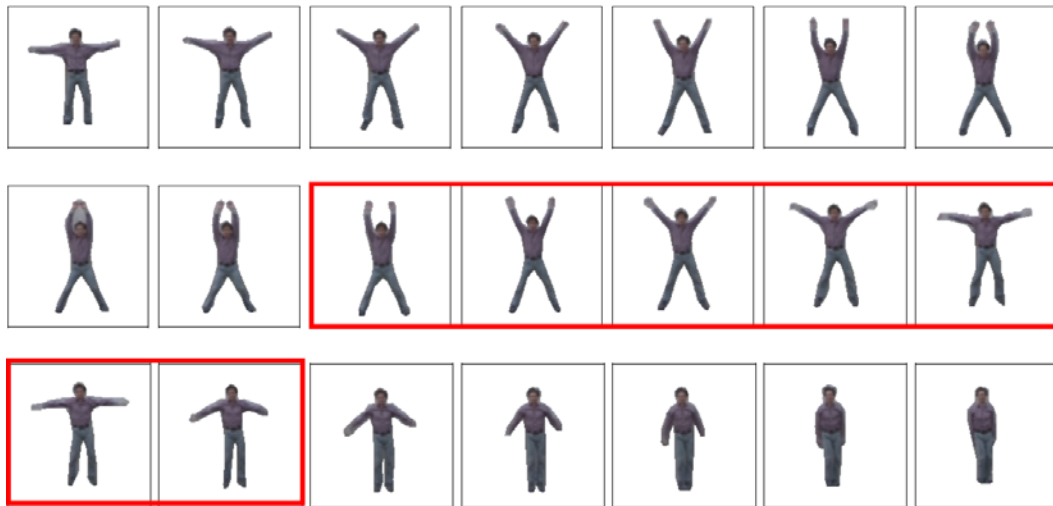
(e)

Figure 3.8 The experiments on test sequence #2: (a) some snapshots of the occluded object in the test sequence; (b) frames reconstructed by Ding *et al.*'s approach; (c) frames reconstructed by Xu *et al.*'s approach; (d) frames reconstructed by the proposed approach; and (e) the inpainting result derived by our approach.

In the third experiment, we used a video sequence (test sequence #3) from the Weizmann database [46] to evaluate our method. We removed 7 of the 55 frames in the sequence. Figure 3.9(a) shows part of the sequence (21 frames). The 7 frames bounded by the red rectangle were the ones removed before the experiment. Figure 3.9(b), (c), and (d) show, respectively, the missing frames reconstructed by the three approaches; and Figure 3.9(e) shows the trajectories reconstructed by the three approaches in the manifold space.

Table 3.3 details the results of the ground-truth method and the three

compared methods. The top row shows the sequence of missing ground-truth postures. The second, third, and fourth rows show the missing frames reconstructed by the two methods in [30] and [39] and our method, respectively. The black parts of the figures are the ground-truth postures; the gray parts are perfectly matched portions; and the red portions belong to reconstructed postures. Note that the first frame reconstructed by Ding *et al.*'s method covers a broad area (the red area above the head). Only this method may generate such results. In terms of the accuracy of the reconstructed frames, our method reconstructed the most accurate postures overall. However, Xu *et al.*'s method reconstructs the most accurate postures in the last of the 7 missing frames. The match rate was 94.3% compared to that of the ground-truth. In contrast, the accuracies of the postures reconstructed by Ding *et al.*'s method and our method are 67.7% and 77.2% respectively compared to that of the ground-truth posture.



(a)



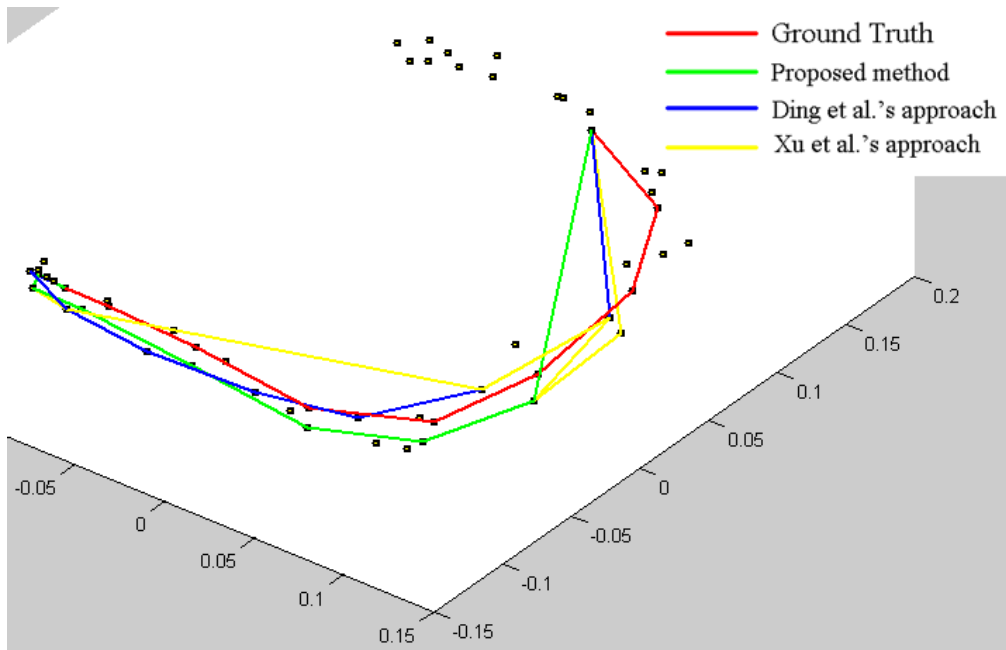
(b)



(c)



(d)










(e)

Figure 3.9 The experiments on test sequence #3: (a) partial sequence of the test sequence in which the red rectangle indicates the 7 missing frames; (b) the frames reconstructed by Ding *et al.*'s approach; (c) the frames reconstructed by Xu *et al.*'s approach; (d) the frames reconstructed by the proposed approach; and (e) the corresponding trajectory information of predicted object motion generated by the compared approaches.

Table 3.3 Comparison of the ground-truth postures and the reconstructed missing postures (The parts in black, red and gray represent the ground-truth postures, reconstructed postures, and perfectly matched portions, respectively)

Ground-truth								Average
Ding <i>et al.</i> [10]								71.3%
	72.7%	76.2%	71.1%	69.2%	72.0%	70.3%	67.7%	
Xu <i>et al.</i> [39]								75.7%
	60.6%	94.0%	68.7%	72.8%	73.3%	66.1%	94.3%	

Ours								80.9%
	83.0%	94.0%	81.3%	73.7%	79.7%	77.5%	77.2%	

3.4 Summary

In this Chapter, we proposed a human object inpainting scheme that divides the process into three steps: human posture synthesis, graphical model construction, and posture sequence estimation. In addition, we also define two constraints on the motion continuity property. The first constraint sets a threshold to confine the maximum search distance; and the second restricts the range of the search direction. With the two constraints, the number of possible candidates between any two consecutive postures can be minimized to a satisfactory extent. We then apply the MRF model to perform global matching. The experiment results demonstrate that the proposed approach outperforms two existing state-of-the-art approaches.

Chapter 4

Object Posture Temporal Super-Resolution Using Tensor Decomposition-Based Manifold Learning

In this Chapter, we describe the proposed framework for Object posture super-resolution using tensor decomposition-based manifold learning. First, we give an introduction about this research topic. The proposed approach is then described. Next, we detail the experiment results. Finally, we present our conclusions.

4.1 Introduction

Super-resolution (SR) [48]–[54] is a class of technique that enhance the resolution of existing images/videos. However, existing SR methods may fail to produce realistic and smooth results while dealing with sequences of human motion. Since human motion usually contains repeated postures, one may insert interpolated postures into the LR input sequence to increase the temporal resolution. In order to generate postures and animate animal/human motion, Xu *et al.* [39] proposed energy minimizing approach to animate motions. However, energy minimization process did not include human motion model, the performance is unstable

and very sensitive to the selected parameters. In [10], Ding *et al.* proposed a rank minimization approach to model and synthesize human motion for video inpainting. This rank minimization approach would usually produce good results as far as the object's motion is periodic. Makihara *et al.* [59] proposed a reconstruction-based method to synthesize periodic human motion with high frame rate from a single periodic motion sequence. Under the constraint of periodic motion, their method could also produce good experiment results.

Nevertheless, since human motion is not always periodic, a single motion sequence could provide only limited and insufficient information to generate high quality temporal SR sequences. Therefore, in this work, we propose using learning-based approach to extract motion tendency from a set of learning sequences and then synthesize interpolated human postures using the learned motion tendency as the prior information. Note that, the extracted motion tendency should preserve only the motion-related information regardless of individual discrepancy in the learning sequence. In [60], Elgammal *et al.* introduced a framework to separate motion data into person and motion factors. However, while we use this decomposed motion factors to increase the temporal resolution of

human motion, we found it difficult to get a stable result. The main reason is because the decomposed person and motion factors are not guaranteed to be orthogonal. Although the multilinear analysis tool like, tensor decomposition, is able to discover the orthogonal factors, the limitation of tensor decomposition is that the motion data need to be arranged into various orthogonal factors beforehand. Such requirement makes it hard if we apply tensor decomposition to decompose motion data into orthogonal factors. Typically, human motion sequences would have different lengths or different sampling rates. In this work, we propose a motion data alignment scheme which can automatically arrange motion data in tensor. Then, we can apply tensor decomposition to decompose motion data into orthogonal factors. Based on decomposed result, we can reconstruct the motion trajectory of LR input sequence. Finally, the global feature, reconstructed motion trajectory and object inpainting which can maintain local motion continuity are combined together to obtain final result.

The proposed framework consists of three steps: graphical model construction, motion trajectory reconstruction and posture selection. The first step, graphical model construction, projects each input motion

sequence into a manifold space and then represent the projected sequence by a motion trajectory. This low-dimensional representation provides a simple and concise representation for human motion. Secondly, we extract the motion and person factors via tensor decomposition, and then use the motion factor extracted from learning sequences and the person factor extracted from the LR input sequence to reconstruct the motion trajectory for the input sequence. Finally, we adopt the human object inpainting technique [61] to select interpolated postures based on the reconstructed motion trajectory.

4.2 Object Posture Temporal Super-Resolution

4.2.1 Overview of the Proposed Method

We propose an object posture super-resolution scheme that can increase the temporal resolution of human motion sequence. Initially, we assume that the objects have been extracted by an automatic object segmentation scheme [19], or by an interactive extraction scheme [20]-[22]. We also assume that the posture number is enough for posture selection based on the observation that human motion usually contains

repeated postures.

Our primary goal is to transfer the motion factor from HR learning sequences to LR input sequence in order to increase the temporal resolution of LR input sequence. Figure 4.1 shows the flowchart of the proposed posture super-resolution scheme which is comprised of three steps: graphical representation of object postures, temporal super-resolution using tensor decomposition-based manifold learning and posture selection. The first step of posture super-resolution involves calculating the similarity value between postures. Then, all postures are projected onto manifold space and we link the postures that appear in adjacent frames in manifold space. After applying the above procedure, we can obtain a graphical representation of the object's motion which provides a simple representation of an object's motion. Next, we extract some significant points along motion trajectory of each learning sequence. These significant points are invariant to different persons and are used to align the motion data of different learning sequences. This process can avoid the affect of different capture rate of cameras and different motion speed. After data alignment, a fixed number of m sampled points is extracted and used to represent the motion trajectory for each training

sequence. As to the LR input sequence alignment, we find k postures (k represents the posture number of LR input sequence) among the m position of tensors and arrange the coordinate value of all postures in tensor. After the above data alignment, we extract the motion factor from only the learning sequence, and extract the person factor form only the column with complete postures. Next, we calculate the value of core tensor using the extracted orthogonal factors and available tensor data. Finally, with the obtained orthogonal factors and the core tensor, we obtain the complete tensor data and then use the tensor data to reconstruct the motion trajectory for the LR input sequence. Finally, the reconstructed motion trajectory and object inpainting are used to select suitable object postures in order to increase the temporal resolution of LR input sequence.

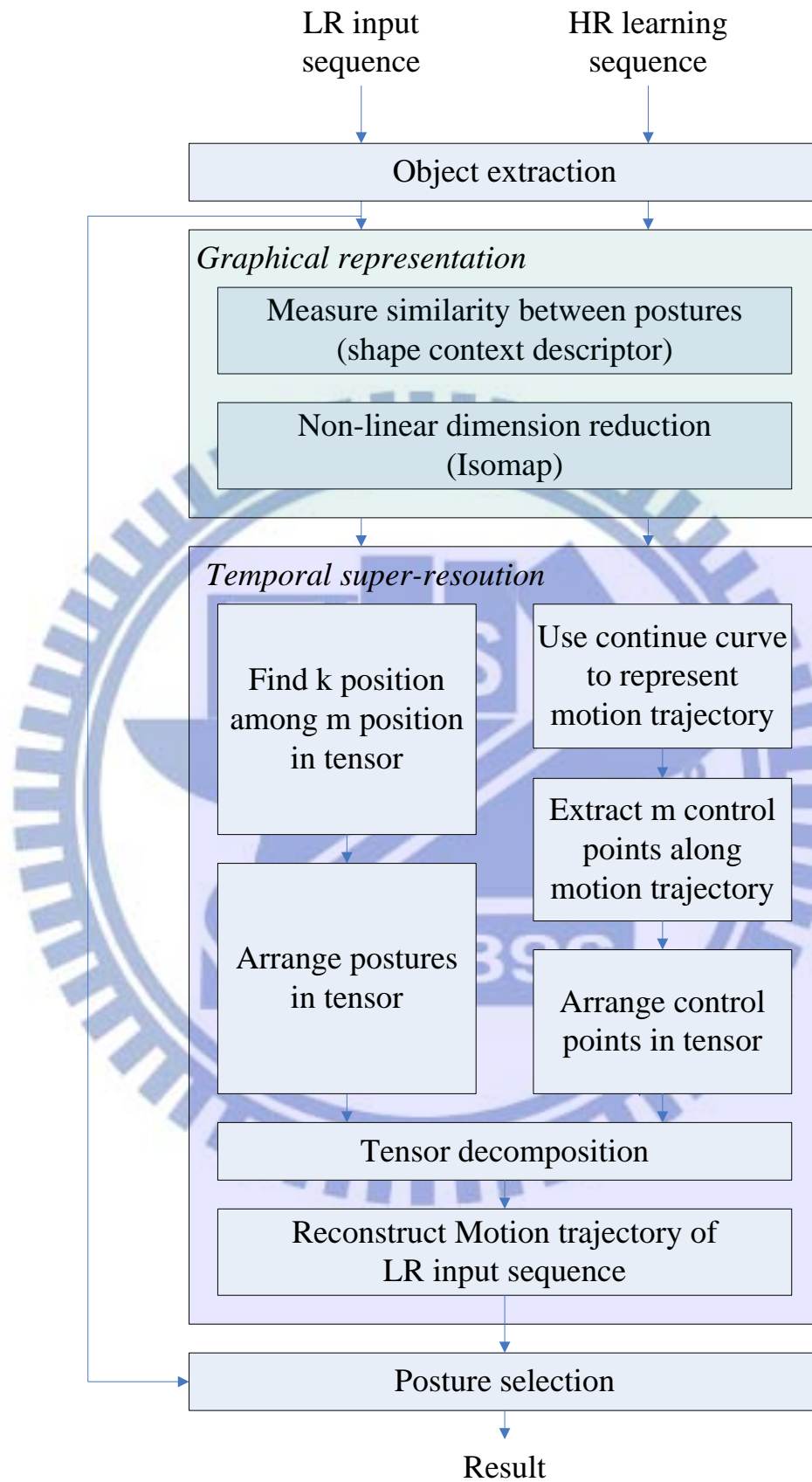


Figure 4.1 Flowchart of the proposed posture super-resolution scheme

4.2.2 Graphical Representation of Object Motion

The graphical representation aims to provide a simple and concise representation of a human motion sequence. To obtain motion trajectory of motion sequence, we utilize our previous Graphical Representation method [47] that was mainly designed for generating motion trajectory of input human motion.

4.2.3 Temporal Super-Resolution Using Tensor Decomposition-based Manifold Learning

After constructing the graphical model, we next wish to transform each human motion sequence into a motion trajectory in the manifold domain. However, since the LR input sequence usually contains poor motion content with low frame rate, its projected motion trajectory in the manifold space would become non-smooth and unreliable. Therefore, we propose to first apply tensor decomposition to separate motion trajectories into two orthogonal factors: motion and person factors. Next, we transfer the motion factor extracted from HR learning sequences to the input sequence and combine the input sequence's person factor to synthesize the motion trajectory for the input sequence with high frame

rate. However, the limitation of tensor decomposition is that the motion data need to be arranged into various orthogonal factors beforehand. Such requirement makes it hard if we apply tensor decomposition to decompose motion data into orthogonal factors. Typically, human motion sequences would have different lengths or different sampling rates. In this work, we propose a motion data alignment scheme which can automatically arrange motion data in tensor. Then, we can apply tensor decomposition to decompose motion data into orthogonal factors.

Tensor is a general form of matrices which defines multi-linear operators over a set of vector spaces and provides a unified mathematical framework for linear analysis. The decomposition of tensor can be seen as a generalization of Singular Value Decomposition (SVD) of matrices. The tensor could be expressed as the product of N -orthogonal spaces as:

$$T = C \times_1 S_1 \times_2 S_2 \times_3 \dots \times_N S_N \quad (4.1)$$

as illustrated in Figure 4.2(a), T denotes the tensor data, C denotes the core tensor, and S_n stands for the n -th orthogonal sub-space.

The tensor decomposition process includes two steps:

1. For $n = 1, 2, \dots, N$, computing the matrix S_n by conducting SVD on the flattened matrix $D_{(n)}$ (as shown in Figure 4.2(b)) and then setting S_n to be the left matrix of the SVD.
2. Finding the core tensor in (4.1).

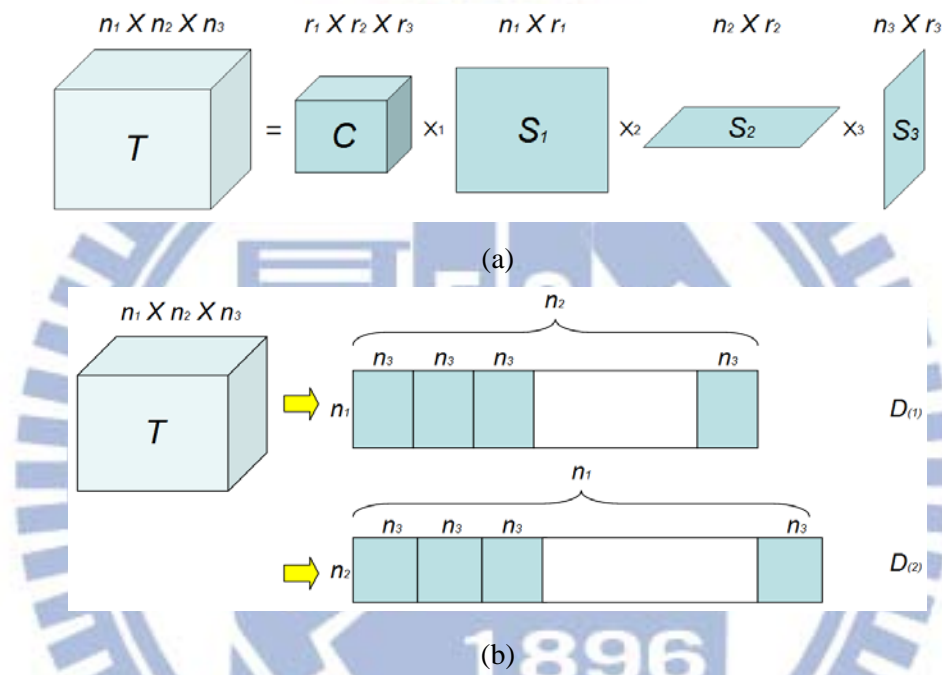


Figure 4.2 Illustration of tensor decomposition and arrangement: (a) a tensor data is decomposed into the product of core tensor and orthogonal factors, and (b) a tensor is flattened in two different ways to obtain flattened matrices.

Before using tensor decomposition to obtain the orthogonal factors, we will need to arrange motion data in the subspaces of tensor in terms of certain attributes. Since human motion sequences contain no definite labels, we need to take special care to correctly organize motion data in tensor. Below we present our proposed motion data alignment method.

We first use a continue motion curve to represent the motion trajectory for each HR learning sequence. Each motion trajectory is

normalized into the same temporal duration and then mapped into a motion curve by polynomial regression. Next, we find some points with significant motion content along the motion trajectory for data alignment. Two examples are shown in Figure 4.3, where each motion trajectory along the first dimension in the manifold domain has some wave crests and troughs. These wave crests and troughs occur just when the person finishes a previous motion and starts to perform the next motion. The other postures in-between the wave crests and troughs would usually contain slow motion due to the human body constraint. These properties as shown in Figure 4.3 are actually invariant to different persons. Therefore, we could sample the points on the wave crests and troughs as the significant points for each motion curve.

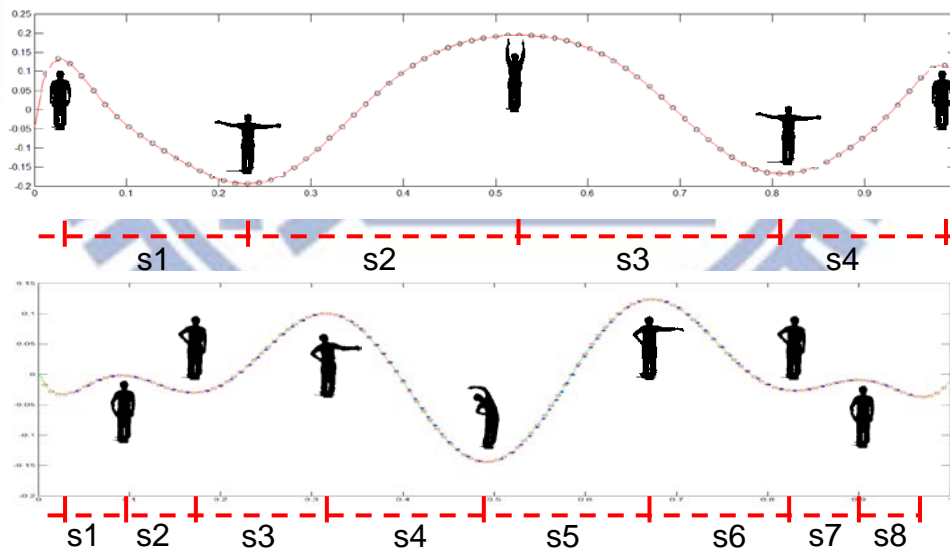


Figure 4.3 Illustration of the low-dimensional manifolds of two different posture sequences and the corresponding postures at the crests and troughs of the manifold.

In addition, to make sure that the sampled points contain sufficient information to represent the original motion trajectory, we additionally

sample n points on the motion curve between every two neighboring key points. These additional points are uniformly sampled under the constant motion assumption between two neighboring key points. The number n is determined by minimizing the distortion between the original motion trajectory and the reconstructed motion trajectory from the sampled points. The threshold is set as the shape context distance between two continuous postures of human motion with static motion. Finally, a fixed number of m sampled points is used to represent the motion trajectory for each training sequence. We then arrange these m points in tensor.

As to the input sequence alignment, since the LR input sequence usually does not contain reliable low-dimensional motion trajectory information, we choose to align the motion data using the raw postures instead of the points along the motion trajectory of test sequence. In order to find k postures among the m sampled points, we arrange the coordinate value of all postures to form a histogram distribution with k bins as shown in Figure 4.4. Then, we find k out of m sampled points along the mean motion trajectory of HR training sequences, where the histogram of k sampled points is similar to the histogram of the input sequence. The similarity between two histogram distributions is calculated by using the Bhattacharyya coefficient as follows:

$$BC(h_{in}, h_{tr}) = \sum_{i=1}^k \sqrt{h_{in}(i)h_{tr}(i)}, \quad (4.2)$$

where h_{in} and h_{tr} respectively represent the histogram of the input LR sequence and the histogram of training sequence.

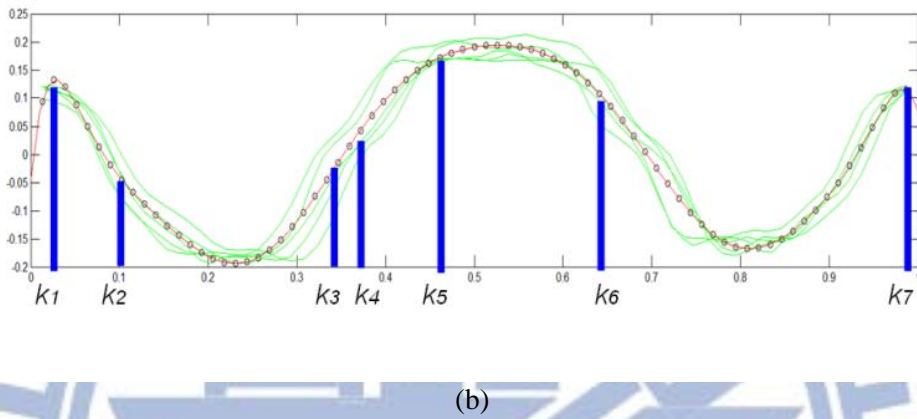
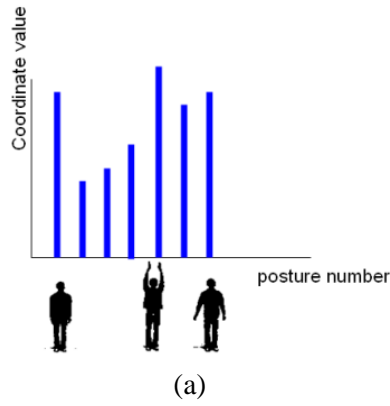


Figure 4.4 (a) The coordinates of the k postures of the LR input sequence. (b) We try to find k reference points among m reference points along the mean motion curve of all the HR learning sequences. The index of the k reference points indicates the suitable position in tensor of the input sequence postures.

After the above data alignment, we now arrange all the sequences including the HR learning and the LR input sequence in tensor. As shown in Figure 4.5, since the tensor is not complete, we cannot directly apply tensor decomposition. Thus, we extract the motion factor from only the learning sequence as indicated by the red rectangle in Figure 4.5, and extract the person factor from only the column with complete postures as indicated by the blue rectangles in Figure 4.5. Next, we calculate the value of core tensor using the extracted orthogonal factors and available

tensor data. Finally, with the obtained orthogonal factors and the core tensor, we obtain the complete tensor data and then use the tensor data to reconstruct the motion trajectory for the input LR sequence.

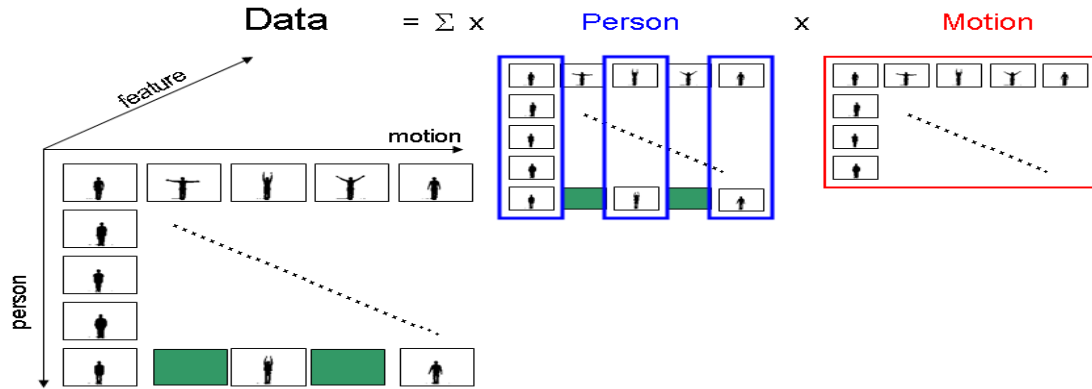


Figure 4.5 Our scheme of arranging training postures into tensor data, where the green rectangles represent unknown object postures in the tensor. In tensor decomposition, we extract the motion factor only from the training sequences as indicated by the red rectangles and the person factor from the columns with complete postures as indicated by the blue rectangles.

4.2.4 Posture Selection

We utilize our previous object inpainting method [47] that was mainly designed for maintaining local motion continuity. In our super-resolution application, we determine the values in the above two constraints based on the reconstructed motion trajectory. The number of upsampled postures p between every two neighboring postures is first specified by the user. After the value of p is determined, we next calculate the possible

positions of the upsampled postures in the manifold space. Once we have the coordinate information of all the upsampled and available postures, we could determine the values in the above two constraints for each local region.

4.3 Experimental Results

To evaluate the effectiveness of the proposed method, we perform experiments on several human object sequences, parts of them were captured with a camcorder and the remaining ones were downloaded from the Weizmann database [46]. In the experiments, we sub-sample each human sequence at different sampling rates to generate the object sequences of low temporal resolutions. Then, we apply the proposed learning-based temporal SR method to synthesize the HR motion sequences. We compare the performance of the proposed method with that of the approaches in [10], [39], [59], [60] and [61]. Due to the space limit, we only show part of the comparison results.



























































In the first experiment, we subsampled test sequence #1 with totally 85 frames under different subsampling rates ranging from 2 to 10. Figure 4.6(a) compares the reconstruction accuracies between the ground-truth










sequence and the reconstructed sequences obtained using the six different approaches for various down-sampling rates. The result shows that the proposed temporal SR method does not only consistently outperform the other methods, but also achieves stably high accuracy of better than 94% under all the nine subsampling rates. Because the proposed motion synthesis method is more effective in extending the frame rate of an object's motion and maintaining motion continuity simultaneously, we conclude that it is more suitable for increasing temporal resolution than the compared methods. On the other hand, the performances of Ding *et al*'s approach [10] and Makihara *et al*'s approach [59] schemes typically degrade as the subsampling rate increases, since the available information for reconstructing HR sequence becomes fewer and less accurate when the temporal resolution of the input LR sequence decreases. The accuracy of Elgammal *et al*'s approach [58] under different sample rate is about 91%. Since, the proposed method can ensure the orthogonal property between decomposed factors. The proposed method can yield better reconstructed result than Elgammal *et al*'s approach [58]. Our previous object inpainting method [60] performs the second best at subsampling rates lower than 8. The motion animation scheme [39] composes a

sequence of smooth posture motion from a set of available postures by executing an energy minimization process. Since the performance of motion animation scheme depends mainly on the two postures at both ends and the available posture database, this scheme can also achieve stable performance under different subsampling rates. However, since it does not take into account the low-dimensional manifold prior of human motion, its performance is significantly lower than the proposed method.

Table 4.1 illustrates the results of the ground-truth and the six compared methods during frame 22 to frame 30. The top row shows some snapshots of four sequences used for training, which are obtained from four different persons taking similar actions. The frame numbers of four learning sequences are 65, 75, 65 and 80. The second row shows the ground-truths of nine missing postures, which are dropped in subsampling. The third to eighth rows show the reconstructed missing frames using the six methods. The reconstruction accuracy of each posture is also indicated under the posture. From these selected postures, it is obvious that the postures reconstructed by our method are consistently better than those derived by the compared methods both subjectively and objectively.

Table 4.1 Comparison of the ground-truth postures and the up-sampled postures obtained by different methods for test sequence #1

Training Sequences									
									
									
									
Ground-Truth									
Xu <i>et al</i> 's approach [39]	 90.1%	 91.1%	 89.8%	 90.4%	 89.7%	 94.4%	 85.3%	 88.5%	 87.0%
Elgammal <i>et al</i> 's approach [58]	 91.5%	 90.8%	 89.7%	 90.0%	 86.6%	 85.4%	 85.6%	 85.6%	 85.4%
Ding <i>et al</i> 's approach [10]	 82.2%	 80.2%	 80.2%	 79.9%	 79.4%	 79.9%	 84.1%	 83.7%	 81.8%
Makihara <i>et al</i> 's approach [59]	 82.4%	 80.5%	 80.0%	 79.2%	 78.5%	 78.3%	 83.7%	 81.6%	 81.8%
Object inpainting	 91.5%	 91.1%	 89.8%	 89.3%	 86.6%	 88.0%	 86.1%	 85.4%	 91.2%





































[60]									
Proposed method	 95.5%	 96.1%	 96.1%	 95.8%	 95.3%	 94.4%	 93.1%	 91.6%	 89.5%
































































In the second experiment, we sample test sequence #2 with totally 135 frames under different subsampling rates and perform six different approaches to reconstruct sequence #2. Figure 4.6 (b) shows the reconstruction accuracies between the ground-truth sequence and the reconstructed sequences obtained using the six different approaches for various down-sampling rates. Among the six approaches, the proposed temporal SR method can achieves better result of better than 94% under all the nine subsampling rates and Elgammal *et al's* approach [58] performs the second best at subsampling rates higher than 5. The high reconstructd accuracy of proposed method and Elgammal *et al's* approach [58] are supported by the global motion tendency extracted from the learning sequences. On the other hand, the performances of Ding *et al's* approach [10] and Makihara *et al's* approach [59] schemes without the support of learning sequence typically degrade as the subsampling rate increases, since the available information for reconstructing HR sequence

becomes fewer and less accurate when the temporal resolution of the input LR sequence decreases.

Table 4.2 illustrates the results of the ground-truth and the six compared methods. The top row shows some snapshots of four sequences used for training, which are obtained from four different persons taking similar actions. The frame numbers of four learning sequences are 125, 110, 120 and 110. The second row shows the ground-truths of nine missing postures, which are dropped in subsampling. The third to eighth rows show the reconstructed missing frames using the six methods. The reconstruction accuracy of each posture is also indicated under the posture. From these selected postures, it is obvious that the postures reconstructed by our method are consistently better than those derived by the compared methods both subjectively and objectively.

Table 4.2 Comparison of the ground-truth postures and the up-sampled postures obtained by different methods for test sequence #2

Training Sequences									
									
									
									

Ground-Truth									
Xu <i>et al</i> 's approach [39]	 95.5%	 95.9%	 96.0%	 91.5%	 91.9%	 94.0%	 93.1%	 93.4%	 91.7%
Elgammal <i>et al</i> 's approach [58]	 90.8%	 91.3%	 90.4%	 92.4%	 92.3%	 93.0%	 93.7%	 92.4%	 94.2%
Ding <i>et al</i> 's approach [10]	 85.9%	 86.1%	 83.5%	 88.6%	 87.7%	 82.9%	 80.8%	 87.5%	 83.5%
Makihara <i>et al</i> 's approach [59]	 90.1%	 89.8%	 85.3%	 86.9%	 83.2%	 88.8%	 89.3%	 89.1%	 85.5%
Object inpainting [60]	 85.8%	 86.6%	 86.2%	 90.1%	 90.5%	 88.8%	 78.9%	 77.3%	 81.2%
Proposed method	 95.5%	 96.0%	 96.0%	 95.8%	 95.4%	 94.0%	 93.1%	 92.4%	 91.7%
































In the third experiment, we used a video sequence from the Weizmann database [46] to evaluate the performance of proposed method.





































We sample test sequence 3 (totally 75 frames) under different subsampling rates and perform six different approaches to reconstruct sequence #3. Figure 4.6 (c) shows the reconstruction accuracies between the ground-truth sequence and the reconstructed sequences obtained using the six different approaches for various down-sampling rates. Due to the poor segmentation result of training and test sequences and the small object size in frame, the reconstruction accuracies of proposed temporal SR method is about 76%. But the proposed method still achieves better result among the six approaches when the subsampling rate higher than 2. The poor segmentation result also affects the performance of other methods. The average reconstructed accuracies of Elgammal *et al*'s approach [58] and Xu *et al*'s approach [39] are 70.0% and 70.3%. The degraded speed of reconstructed accuracy about Ding *et al*'s approach [10] and Makihara *et al*'s approach [59] is faster than test sequence #1 and test sequence #3 because of the poor segmentation result.

Table 4.3 illustrates the results of the ground-truth and the six compared methods. The top row shows some snapshots of four sequences used for training, which are obtained from four different persons taking similar actions. The frame numbers of four learning sequences are 65, 75,

73 and 78. The second row shows the ground-truths of nine missing postures, which are dropped in subsampling. The third to eighth rows show the reconstructed missing frames using the six methods. The reconstruction accuracy of each posture is also indicated under the posture. From these selected postures, it is obvious that the postures reconstructed by our method are consistently better than those derived by the compared methods both subjectively and objectively.

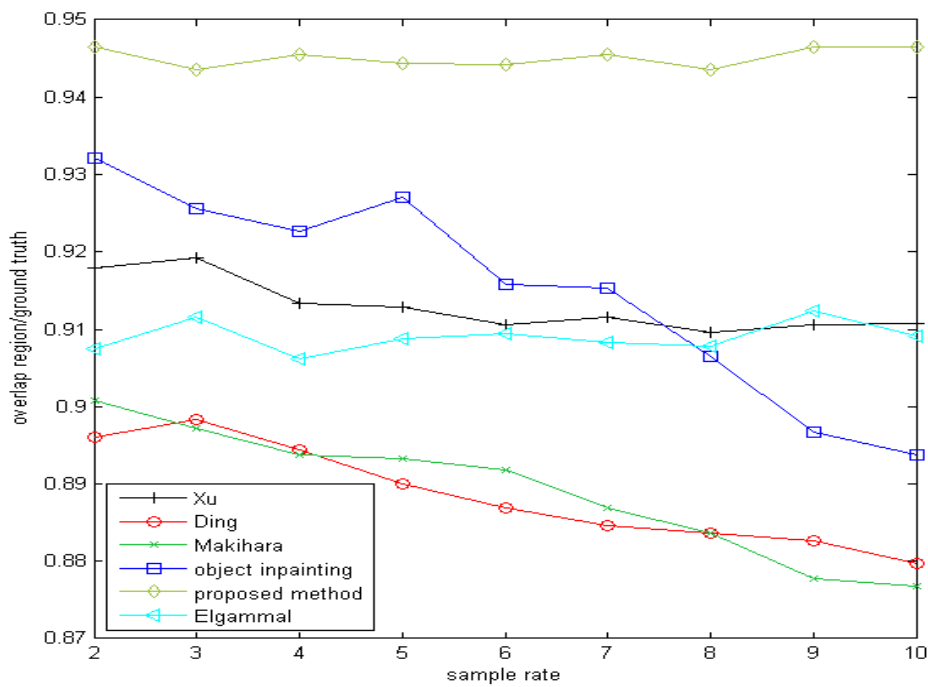
Table 4.3 Comparison of the ground-truth postures and the up-sampled postures obtained by different methods for test sequence #3

Training Sequences									
									
									
									
Ground-Truth									
Xu <i>et al</i> 's approach [39]									
	80.2%	72.2%	53.7%	47.2%	45.4%	49.7%	50.7%	61.4%	66.4%
Elgammal <i>et al</i> 's approach [58]									
	69.0%	73.6%	76.7%	67.0%	67.4%	65.7%	90.3%	80.5%	66.1%

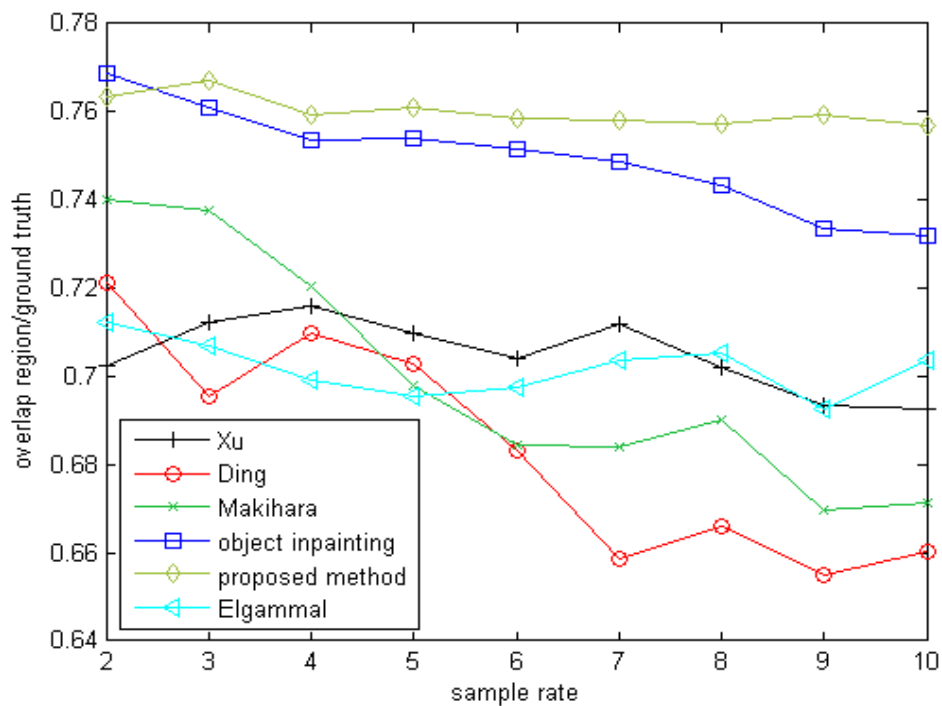
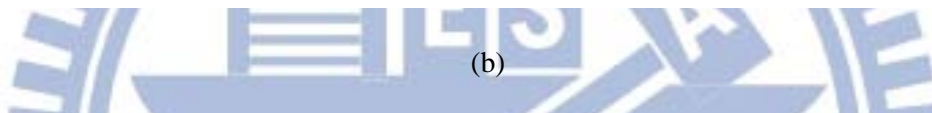
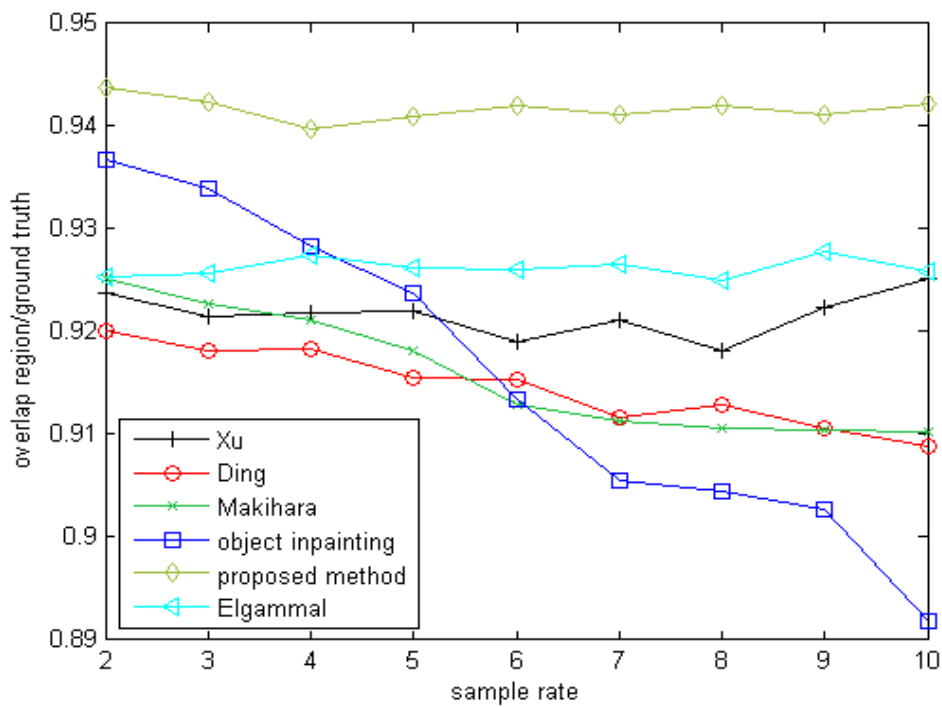
Ding <i>et al</i> 's approach [10]	 83.2%	 78.1%	 75.2%	 69.7%	 81.2%	 72.3%	 89.2%	 76.5%	 83.2%
Makihara <i>et al</i> 's approach [59]	 86.3%	 71.7%	 83.0%	 69.4%	 70.0%	 75.3%	 68.8%	 74.4%	 82.0%
Object inpainting [60]	 72.9%	 71.7%	 78.9%	 69.4%	 81.2%	 77.5%	 75.4%	 65.6%	 77.1%
Proposed method	 75.7%	 85.2%	 71.6%	 73.6%	 74.1%	 72.4%	 73.5%	 71.7%	 68.1%

Note, in the experiments, we compare the performance between the proposed method and our previous object inpainting method [60]. The difference between these two approaches is that the proposed method reconstructs postures based on the rich information in the low-dimensional manifold motion priors learned from the HR training sequences, whereas the object inpainting method [61] utilizes self-contained information in the available LR postures to reconstruct postures without the support of HR training sequences. Since the object inpainting method does not need any HR training sequence, it can be

regarded as the baseline mode of the proposed method that can achieve reasonable reconstruction accuracy without the need of HR training sequences. When HR training sequences are available, as an advanced tool, the proposed tensor decomposition based on manifold learning can significantly improve the accuracy and stability of reconstructed HR postures.



(a)



(c)

Figure 4.6 Comparison of reconstruction accuracies with respect to the ground-truth sequence with nine subsampling rates for test sequence #1. The five

compared methods include Xu *et al*'s approach [39], Ding *et al*'s approach [10], Makihara *et al*'s approach [59], object inpainting [60] and the proposed temporal SR approach.

4.4 Summary

We proposed a human motion temporal super-resolution method which consists of three steps: (1) graphical models construction; (2) motion trajectory reconstruction; and (3) posture selection. In addition, we also proposed a motion data alignment method to correctly arrange motion data from different persons in a tensor so as to increase the accuracy of tensor decomposition. The tensor decomposition effectively decomposes the motion data into two orthogonal factors. With the orthogonal motion and person factors, we transfer the motion factor extracted from training sequences to reconstruct the motion trajectory for the input sequence. Finally, we adopt an object inpainting method on the reconstructed motion trajectory to select interpolated postures. Both global motion tendency and local motion continuity are well preserved in the resultant HR sequence. The experiment results also demonstrate that the proposed approach outperforms two existing state-of-the-art approaches.

Chapter 5

Conclusions and Future Work

5.1 Conclusion

In this dissertation, we have presented two different kinds of object-based video inpainting schemes and a learning-based approach to enrich the content of human motion. First, an object inpainting method based on virtual contour construction was discussed in Chapter 2. Second, we propose an object inpainting method using manifold learning-based posture sequence in Chapter 3. In Chapter 4, we presented a posture temporal super-resolution method using tensor decomposition-based manifold learning.

In Chapter 2, we have proposed a novel method that treats the completion of objects and completion of the background separately. The method is comprised of three steps: virtual contour construction, key posture-based sequence retrieval, and synthetic posture generation. We have also proposed an efficient posture mapping method that uses key posture selection, indexing, and coding operations to convert the posture sequence retrieval problem into a substring matching problem. In addition,

we have developed a synthetic posture generation scheme that enhances the variety of postures available in the database. Our experiment results show that the proposed method generates completed objects with good subjective quality in terms of the objects' spatial consistency and temporal motion continuity. The proposed method still has a few constraints. First, if an object moves nonlinearly during an occlusion period, the virtual contour construction may not compose sufficiently accurate postures. But should there be enough non-occluded portion of the object, the linear motion constraint may be relaxed. Second, currently the proposed method does not deal with the illumination change problem that occurs if lighting is not uniform across the scene. Third, the synthetic posture generation method can only deal with objects that can be explicitly decomposed into constituent components (e.g., a walking person), but may not synthesize complex postures.

In Chapter 3, we have proposed a human object inpainting scheme that divides the process into three steps: human posture synthesis, graphical model construction, and posture sequence estimation. In addition, we define two constraints on the motion continuity property. The first constraint sets a threshold to limit the maximum search distance;

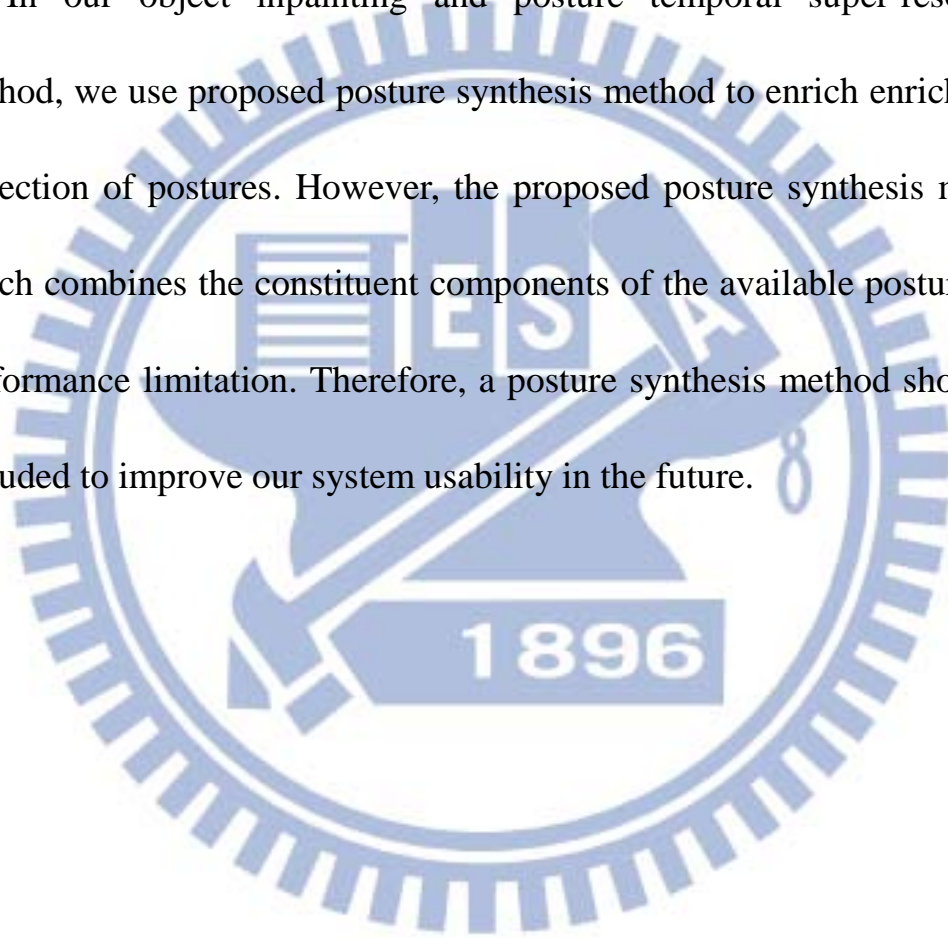
and the second confines the range of the search direction. With the two constraints, the number of possible candidates between any two consecutive postures can be reduced significantly. We then apply the MRF model to perform global matching. The experiment results demonstrate that the proposed approach outperforms two existing state-of-the-art approaches.

In Chapter 4, we proposed a human motion temporal super-resolution method which consists of three steps: (1) graphical models construction; (2) motion trajectory reconstruction; and (3) posture selection. In addition, we also proposed a motion data alignment method to correctly arrange motion data from different persons in a tensor so as to increase the accuracy of tensor decomposition. The tensor decomposition effectively decomposes the motion data into two orthogonal factors. With the orthogonal motion and person factors, we transfer the motion factor extracted from training sequences to reconstruct the motion trajectory for the input sequence. Finally, we adopt an object inpainting method on the reconstructed motion trajectory to select interpolated postures. Both global motion tendency and local motion continuity are well preserved in the resultant HR sequence. The experiment results also demonstrate that

the proposed approach outperforms two existing state-of-the-art approaches.

5.2 Future Work

In our object inpainting and posture temporal super-resolution method, we use proposed posture synthesis method to enrich enriches the collection of postures. However, the proposed posture synthesis method which combines the constituent components of the available postures has performance limitation. Therefore, a posture synthesis method should be included to improve our system usability in the future.



Reference

- [1] K. A. Patwardhan, G. Sapiro, and M. Bertalmío, "Video inpainting under constrained camera motion," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 545–553, Feb. 2007.
- [2] Y. Shen, F. Lu, X. Cao, and H. Foroosh, "Video completion for perspective camera under constrained motion," in *Proc. IEEE Conf. Pattern Recognit.*, pp. 63–66, Aug. 2006, Hong Kong, China.
- [3] Y. T. Jia, S. M. Hu, and R. R. Martin, "Video completion using tracking and fragment merging," *Visual Comput.*, vol. 21, no. 8–10, pp. 601–610, Aug. 2005.
- [4] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 1–14, Mar. 2007.
- [5] V. Cheung, B. J. Frey, and N. Jojic, "Video epitomes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 42–49, June 2005, San Diego, CA.
- [6] T. K. Shih, N. C. Tang, and J.-N. Hwang, "Exemplar-based video inpainting without ghost shadow artifacts by maintaining temporal continuity," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 347–360,

Mar. 2009.

[7] S.-C. S. Cheung, J. Zhao and M. V. Venkatesh, “Efficient object-based video inpainting,” in *Proc. IEEE Conf. Image Process.*, pp. 705–708, Oct. 2006, Atlanta, GA.

[8] J. Jia, Y.-W. Tai, T.-P. Wu, and C.-K. Tang, “Video repairing under variable illumination using cyclic motions,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 832–839, May 2006.

[9] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, “Navier-stokes, fluid dynamics, and image and video inpainting,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 355–362, Dec. 2001, Hawaii.

[10] T. Ding, M. Sznajder, and O. I. Camps, “A rank minimization approach to video inpainting,” in *Proc. IEEE Conf. Comput. Vis.*, pp. 1–8, Oct. 2007, Rio de Janeiro, Brazil.

[11] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, “Full frame video stabilization with motion inpainting,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1150–1163, July 2006.

[12] Venkatesh, M. V., S.-C. Cheung, J. Paruchuri, J. Zhao, and T. Nguyen, “Protecting and Managing Privacy Information in Video Surveillance

System,” in *Protecting Privacy in Video Surveillance*, edited by Andrew Senior, Springer, 2009.

[13] A. Efros and T. Leung, “Texture synthesis by non-parametric sampling,” in *Proc. IEEE Conf. Comput. Vis.*, vol. 2, pp. 1033–1038, 1999.

[14] L. Wei and M. Levoy, “Fast texture synthesis using tree structured vector quantization,” *ACM SIGGRAPH*, pp. 479–488, 2000.

[15] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpainting,” *ACM SIGGRAPH*, pp. 417–424, 2000.

[16] A. Criminisi, P. Perez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting,” *IEEE Trans. Image Process.*, vol.13, no.9, pp. 1200–1212, Sept. 2004.

[17] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum, “Image completion with structure propagation,” *ACM SIGGRAPH*, pp. 861–868, 2005.

[18] T. Huang, S. Chen, J. Liu, and X. Tang, “Image inpainting by global structure and texture propagation,” in *Proc. ACM Conf. Multimedia*, pp. 517–520, Sept. 2007, Augsburg, Germany.

[19] I. Haritaoglu, D. Harwood, and L.S. Davis. “W4: Who? When? Where? What? A real-time system for detecting and tracking people,”

- in Proc. *IEEE Int. Conf. Automatic Face Gesture Recognit.*, pp. 222–227, 1998, Los Alamitos, CA.
- [20] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, “Lazy snapping,” *ACM Trans. Graphics*, vol. 23, no. 3, pp. 303–308, Aug. 2004.
- [21] Y. Li, J. Sun, and H.-Y. Shum, “Video object cut and paste,” *ACM Trans. Graphics*, vol. 24, no.3, pp. 595–600, Aug. 2005.
- [22] L. Yatziv and G. Sapiro, “Fast image and video colorization using chrominance blending,” *IEEE Trans. Image Process.*, vol. 15, no. 5, pp. 1120–1129, May, 2006.
- [23] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.
- [24] Y.-M. Liang, S.-W. Shih, C.-C. A. Shih, H.-Y. M. Liao, and C.-C. Lin, “Learning atomic human actions using variable-length Markov models,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 1, pp. 268–280, Jan. 2009.
- [25] A. Elgammal and C.-S. Lee, “Inferring 3D body pose from silhouettes using activity manifold learning,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 681–688, June 2004. Washington,

DC.

- [26]T. H. Corman, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd Ed., The MIT Press, 2001.
- [27]D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.
- [28]X. Bai, L. J. Latecki, and W.-Y. Liu, “Skeleton pruning by contour partitioning with discrete curve evolution,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 449–462, Mar. 2007.
- [29]D. Cremers, S. J. Osher, and S. Soatto., “Kernel density estimation and intrinsic alignment for shape priors in level set segmentation,” *Int. J. Comput. Vis.*, vol. 69, no. 3, pp. 335–351, Sept., 2006.
- [30]C.-H. Ling, C.-W. Lin, C.-W. Su, H.-Y. M. Liao, and Y.-S. Chen, “Virtual contour-guided video object inpainting using posture mapping and retrieval,” *IEEE Trans. Multimedia*, vol. 13, no. 2, pp. 292–302, Apr. 2011.
- [31]S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 23239–2326, Dec. 2000.

- [32]L. Wang, W. Hu, and T. Tan, “Recent developments in human motion analysis,” *Pattern Recognit.*, vol. 36, no. 3, pp. 585–601, Mar. 2003.
- [33]T. B. Moeslund, A. Hilton, and V. Krüger, “A survey of advances in vision-based human motion capture and analysis,” *Comput. Vis. Image Und.* vol. 104, no. 2–3, pp. 90–126, Nov.–Dec. 2006.
- [34]R. Poppe, “Video-based human motion analysis: An overview,” *Comput. Vis. Image Und.* vol. 108, no. 1–2, pp. 4–18, Oct.–Nov.2007.
- [35]F. Caillette, A. Galata, and T. Howard, “Real-time 3-D human body tracking using variable length Markov models,” in *Proc. British Mach. Vis. Conf.*, pp. 469–478, Sept. 2005, Oxford, United Kindom.
- [36]A. M. Elgammal and C.-S. Lee, “Inferring 3D body pose from sillouettes using activity manifold learning,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 681–688, June 2004, Washington, DC.
- [37]K. Grauman, S. L. Martin, A. Hertzmann, and Z. Popovic, “Style-based inverse kinematics,” *ACM Trans. Graphics*, vol. 23, no. 3, pp. 522–531, Dec. 2004.

- [38]Y. W. Teh and S. T. Roweis, “Automatic alignment of local representation,” in *Proc. Adv. Neural Information Process. Syst.*, vol. 15, pp. 841–848, Dec. 2002, Vancouver, Canada.
- [39]X. Xu, L. Wan, X. Liu, T.-T. Wong, L. Wang, and C.-S. Leung, “Animating animal motion from still,” *ACM Trans. Graphics*, vol. 27, no. 5, Dec. 2008.
- [40]J. K. Aggarwal and Q. Cai, “Human motion analysis: A review,” in *Proc. Nonrigid and Articulated Motion Workshop*, pp. 90–102, June 1997.
- [41]D. M. Gavrilu, “The visual analysis of human movement: A survey,” *Comput. Vis. Image Und.*, vol. 73, no. 1, pp. 82–98, Jan. 1999.
- [42]G. Mori, X. Ren, A. A. Efros, and J. Malik, “Recovering human body configuration: Combining segmentation and recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 206–213, June 2006, New York, NY.
- [43]D. Ramanan and C. Sminchisescu, “Training deformable models for localizations,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 326–333, June 2004, Washington, DC.

- [44]J. B. Tenenbaum, V. de Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.
- [45]I. Borg and P. Groenen, *Modern Multidimensional Scaling: Theory and Applications*, 2nd Ed. Springer-Verlag New York, 2005.
- [46]L. Gorelick, M. Blank, E. Shechtman, M. Irani and R. Basri, “Actions as space-time shapes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007.
- [47]C.-H. Ling, Y.-M. Liang, C.-W. Lin, Y.-S. Chen, and H.-Y. Mark Liao, “Video object inpainting using manifold-based action prediction,” in *Proc. IEEE Conf. Image Process.*, Sept. 2010, Hong Kong, China.
- [48]V. Caselles, J.-M. Morel, and C. Sbert, "An axiomatic approach to image interpolation," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 376–386, Mar. 1998
- [49]X. Li and M. T. Orchard, “New edge-directed interpolation,” *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521–1527, Oct. 2001.
- [50]S. Baker and T. Kanade, “Limits on super-resolution and how to break them,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1167–1183, Sept. 2002.

- [51]Z. Lin and H.-Y. Shum, “Fundamental limits of reconstruction-based super resolution algorithms under local translation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 83–97, Jan. 2004.
- [52]W. T. Freeman, T. R. Jones and E. C. Pasztor, “Example-based super-resolution,” *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, Mar. 2002.
- [53]W. T. Freeman, E. C. Pasztor and O. T. Carmichael, “Learning low-level vision,” *Int. J. Comput. Vis.*, vol. 20, no. 1, pp. 25–47, 2000.
- [54]E. Shechtman, Y. Caspi and M. Irani, "Space-time super-resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 531–544, Apr. 2005.
- [55]R. Souvenir and J. Babbs, “Learning the viewpoint manifold for action recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, Alaska, USA, May 2008, pp. 1–7.
- [56]C.-S. Lee and A. Elgammal, "Modeling view and posture manifolds for tracking," in *Proc. IEEE Conf. Comput. Vis.*, Rio de Janeiro, Brazil, Oct. 2007, pp. 1–8.

- [57]X. Zhang and G. Fan, “Joint gait-pose manifold for video-based human motion estimation,” in *Proc. IEEE Comput. Vis. Pattern Recognit. Workshops*, Colorado Springs, USA, June 2011, pp. 47–54.
- [58]A. Elgammal and C.-S. Lee, “Nonlinear manifold learning for dynamic shape and dynamic appearance,” *Comput. Vis. Image Und.*, vol. 106, no. 1, pp. 31–46, Apr. 2007.
- [59]Y. Makihara, A. Mori and Y. Yagi, “Temporal super resolution from a single quasi-periodic image sequence based on phase registration,” in *Proc. Asian Conf. Comput. Vis.*, Queenstown, New Zealand, Nov. 2010, pp. 107–120.
- [60]A. Elgammal, and C.-S. Lee, “Separating style and content on a nonlinear manifold,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Washington DC, USA, June, 2004. pp. 478–485.
- [61]C.-H. Ling, Y.-M. Liang, C.-W. Lin, Y.-S. Chen, and H.-Y. M. Liao, “Human object inpainting using manifold learning-based posture sequence estimation,” *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3124–3135, Nov. 2011.

Publication List

- [1] **Chih-Hung Ling**, Yu-Ming Liang, Chia-Wen Lin, Hong-Yuan Mark Liao, and Yong-Sheng Chen, “Human object inpainting using manifold learning-based posture sequence estimation,” *IEEE Trans. Image Processing*, vol. 20, no. 11, pp. 3124–3135, Nov. 2011.
- [2] **Chih-Hung Ling**, Chia-Wen Lin, Chih-Wen Su, Yong-Sheng Chen, and Hong-Yuan Mark Liao, “Virtual contour guided video object inpainting using posture mapping and retrieval,” *IEEE Trans. Multimedia*, vol. 13, no. 2, 11 pages, April 2011.
- [3] **Chih-Hung Ling**, Yu-Ming Liang, Chia-Wen Lin, Hong-Yuan Mark Liao, and Yong-Sheng Chen, “Video object inpainting using manifold-based posture estimation,” *IEEE Int. Conf. Image Processing*, Sept. 2010, Hong Kong, China.
- [4] **Chih-Hung Ling**, Chia-Wen Lin, Chih-Wen Su, Hong-Yuan Mark Liao, and Yong-Sheng Chen, “Video object inpainting using posture mapping,” in *Proc. IEEE Int. Conf. Image Processing*, Nov. 2009, Cairo, Egypt.