3D Pedestrian Tracking and Virtual Reconstruction of Ceramic Vessels

Using Geometric and Color Cues

A Thesis

Submitted to the Faculty

of

Drexel University

by

Zhongchuan Zhang

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy

April 2016

Dedications

This thesis is dedicated to my wife Jingying Hu and our parents, for their endless love and support.

Acknowledgements

I would like to express my sincere gratitude to all the people who helped me during my Ph.D. program at Drexel University.

First of all, I would like to thank my advisor, Dr. Fernand Cohen, for supporting me during these past six years. He is the funniest advisor and one of the smartest people I know. I hope that I could be as lively, enthusiastic, and energetic as him. He has provided insightful discussions about the research. Without his help and guidance, I could not have a productive graduate life.

I would like to thank Dr. Nagarajan Kandasamy, Dr. Ioannis Savidis, Dr. David Breen and Dr. Tom Chmielewski, for serving on my thesis defense committee. I have learned a lot from their valuable suggestions and encouragement during my research work.

I would also like to thank the alumni and students in our lab, Dr. Ezgi Taslidere and Zexi Liu, for their help and all the fun we have had during the Ph.D. study.

Lastly and most importantly, I would like to express my greatest appreciation to my beloved wife Jingying Hu for her endless love, encouragement and care, and to our parents for their continuous support.

TABLE OF CONTENTS

LIST OF TABLES vi

LIST OF FIGURES vii

ABSTRACT xi

1. INTRODUCTION 1

1.1 Problems and Motivation in Pedestrian Tracking 1

1.2 Problems and Motivation in Virtual Reconstruction of Broken Vessels . 4

1.3 Geometric and Color Cues 7

1.4 Contributions of the Thesis 10

1.5 Organization of the Thesis 11

2. TRACKING PEDESTRAINS IN 3D 13

2.1 Introduction 13

2.2 Related Work 16

2.3 3D Pedestrian Tracking in Uncrowded Scenes 20

2.3.1 Background Subtraction 21

2.3.2 Potential Head Top Segment Detection 23

2.3.3 3D Head Position Estimation 25

2.3.3.1 Establishing Disparities along the Potential Head Top Segment 26

2.3.3.2 Computing 3D Head Coordinates 27

2.3.4 Pedestrian Tracking 29

2.3.5 Experiments 30

2.3.5.1 Experiment Setup 30

2.3.5.2 Experiment Results 31

2.4 3D Pedestrian Tracking in Crowded Scenes 41

2.4.1 Head Area Existence Probability Calculation 43

2.4.2 Potential Head Top Segment Detection 50

2.4.3 Pedestrian Tracking 53

2.4.3.1 Tracking Based on the Color Cues of a Head Top 53

2.4.3.2 Tracking by Fusing the Geometric and Color Cues 57

2.4.4 Experiments 61

2.4.4.1 Experiment Setup 61

2.4.4.2 Experiment Results 62

2.5 Conclusions 68

3. SCHEDULING PTZ CAMERAS FOR FACE IMAGE CAPTURE BASED ON 3D PEDESTRIAN TRACKS 70

3.1 Introduction 70

3.2 Best PTZ Camera Selection based on Capture Quality 74

3.2.1 Capture Quality Measures 74

3.2.1.1 Head Visibility 74

3.2.1.2 View Angle of the Frontal Face 76

3.2.1.3 Camera-target Distance 77

3.2.1.4 Mechanical Limits 77

3.2.2 PTZ Camera Selection 78

3.3 Experiments 80

3.4 Conclusions 85

4. VIRTUAL RECONSTRUCTION OF BROKEN CERAMIC VESSELS 86

4.1 Introduction 86

4.2 Generic Models 93

4.3 Modeling Fragments and Generic Models based on their Color Markings 94

4.4 Weighted Moments 97

4.5 Establishing Matching Markings based on their Geometric Relation.... 97

4.5.1 Choice of the S-weighted Function 98

4.5.2 Absolute Invariants for Marking Matching 100

4.6 Estimating the Geometric Transformation between the Matching Markings 103

4.7 Aligning Fragments against Generic Models 104

4.8 Experiments 105

4.8.1 Data Collection 105

4.8.2 Alignment Results 107

4.9 Conclusions 112

5. CONCLUSIONS AND FUTURE WORK 113

5.1 Conclusions 113

5.2 Future Directions 114

LIST OF REFERENCES 117

VITA 127

LIST OF TABLES

Table 2.1: The errors of the estimated tracks on the planar ground 36

Table 2.2: The errors of the estimated tracks in the non-planar ground 36

Table 2.3: The overall tracking errors..... 37

Table 2.4: The smallest and largest four average errors of the estimated tracks when only the geometric cues are used 65

Table 2.5: The smallest and largest four average errors of the estimated tracks when both the geometric and color cues are used 65

Table 2.6: The overall tracking errors when only the geometric cues are used.... 65

Table 2.7: The overall tracking errors when both the geometric and color cues are used 66

Table 4.1: Reconstruction results and errors..... 110

LIST OF FIGURES

Figure 1.1: Tracking a basketball player by consistently labelling him across frames 1

Figure 1.2: Ground plane trajectories of the basketball players 3

Figure 1.3: Mending ceramic fragments manually 5

Figure 1.4: Some applications based on geometric cues 8

Figure 1.5: Some applications based on color cues 9

Figure 2.1: Occlusions under a side and overhead view of the same scene 14

Figure 2.2: A crowded scene from an overhead view 15

Figure 2.3: 3D template matching 19

Figure 2.4: Perspective height field 20

Figure 2.5: Flowchart of pedestrian tracking under uncrowded scenes..... 20

Figure 2.6: Extracted foreground blobs in a frame from the left camera 22

Figure 2.7: Detection of the potential head top segment using projective geometry 24

Figure 2.8: The centroid and the furthest point of a person..... 25

Figure 2.9: Finding the corresponding pixel from the right image (b) for a pixel on the potential head top segment in the left image (a) 26

Figure 2.10: 3D head position calculation 28

Figure 2.11: The virtual scenes of the train station concourse 30

Figure 2.12: The frames captured by two cameras with people walking on the planar ground 32

Figure 2.13: The frames captured by two cameras with people walking on non-planar ground 32

Figure 2.14: X-Y plane tracking results using two different methods when people walking on the planar ground. 34

Figure 2.15: X-Y plane tracking results using two different methods when people walking on the non-planar ground 35

Figure 2.16: A part of the frames showing (a) A not well detected potential head top segment and (b) A not well detected head point. The white segment denotes the potential head top segment, and the green and red dots denotes the foreground blob center and detected head point respectively..... 38

Figure 2.17: The close-up facial images captured in the scene with planar ground when using our method ((a)-(c)) and BC method ((d)-(e)), respectively. 39

Figure 2.18: The close-up facial images captured in the scene with non-planar ground by using the 3D head points detected by our approach. 39

Figure 2.19: Potential head top segment detected using the method for uncrowded scenes 41

Figure 2.20: Flowchart of Pedestrian tracking in crowded scenes 42

Figure 2.21: Projective geometry of a camera 43

Figure 2.22: Part of the image captured by the left camera as shown in Figure 2.23(a)

..... 44

Figure 2.23: Polar mapping..... 46

Figure 2.24: The reference projections and heights for foreground pixels..... 47

Figure 2.25: Pre-computing the height and width of the reference projection for a pixel in the polar mapped image 48

Figure 2.26: ROIs with color coded head area existence probabilities in the polar mapped image 50

Figure 2.27: Establishing a potential head top segment 51

Figure 2.28: A head point in the current frame (b) is detected using the projective geometry and associated with the head point in the previous frame (a) based on the constant velocity within 2 consecutive frames. The frames in (a) and (b) are from the left overhead camera 57

Figure 2.29: A virtual train station concourse 61

Figure 2.30: Two frames captured by the left camera with the head points detected by combining the geometric and color cues 63

Figure 2.31: The ground plane tracking results 64

Figure 2.32: Close-up facial images of person 3 captured by the PTZ camera 68

Figure 3.1: A camera network for close-up face image capture 73

Figure 3.2: Evaluating the head visibility of a person of interest from the view of a PTZ camera 75

Figure 3.3: An overhead view of a person walking on the ground 76

Figure 3.4: Selecting a most appropriate PTZ camera to capture frontal face images 79

Figure 3.5: PTZ camera distribution 80

Figure 3.6: Close-up face image capturing results when the handoff success probability is not considered 81

Figure 3.7: The face image captured at frame 108 when the handoff success probability is considered 82

Figure 3.8: 4 face image capturing locations on the estimated track of person 1. 83

Figure 3.9: Face images in (a) and (b) are captured when person 1 is at the positions shown as the red circles in (c) and (d) respectively 84

Figure 4.1: Ceramic fragments with eroded edges 90

Figure 4.2: Ceramic fragments 91

Figure 4.3: Aligning a fragment to a corresponding generic model based on color markings 92

Figure 4.4: A 3D point set and its convex hull where the red asterisks denote the vertices and the blue dots denote the rest of the points inside the convex hull 95

Figure 4.5: Geometric interpretation of equation (4.7). On the left there are four 3D points forming a tetrahedron, and on the right there are the corresponding points and tetrahedron after an affine transformation T_A. 99

Figure 4.6: 3D scanner and scan setup 106

Figure 4.7: 3D scanned fragments from the vessels with marking on them..... 107

Figure 4.8: Some of generic models for different types of vessels..... 108

Figure 4.9: Alignment results where fragments with red boundaries are aligned using our method and those with green boundaries are aligned manually 109

Figure 4.10: Fragments without any markings on them 110

Figure 4.11: Two misaligned fragments 111

Figure 4.12: Two vessels without any color markings 111

Abstract

3D Pedestrian Tracking and Virtual Reconstruction of Ceramic Vessels Using Geometric and Color Cues
Zhongchuan Zhang

Object tracking using cameras has many applications ranging from monitoring children and the elderly, to behavior analysis, entertainment, and homeland security. This thesis concentrates on the problem of tracking person(s) of interest in crowded scenes (e.g., airports, train stations, malls, etc.), rendering their locations in time and space along with high quality close-up images of the person for recognition. The tracking is achieved using a combination of overhead cameras for 3D tracking and a network of pan-tilt-zoom (PTZ) cameras to obtain close-up frontal face images. Based on projective geometry, the overhead cameras track people using salient and easily computable feature points such as head points. When the obtained head point is not accurate enough, the color information of the head tops across subsequent frames is integrated to detect and track people. To capture the best frontal face images of a target across time, a PTZ camera scheduling is proposed, where the 'best' PTZ camera is selected based on the capture quality (as close as possible to frontal view) and handoff success (response time needed by the newly selected camera to move from current to desired state) probabilities. The experiments show the 3D tracking errors are very small (less than 5 cm with 14 people crowding an area of around 4 m^2) and the frontal face images are captured effectively with most of them centering in the frames.

Computational archaeology is becoming a success story of applying computational tools in the reconstruction of vessels obtained from digs, freeing the expert from hours of intensive labor in manually stitching shards into meaningful vessels. In this thesis, we concentrate on the use of geometric and color information of the fragments for 3D virtual reconstruction of broken ceramic vessels. Generic models generated by the experts as a rendition of what the original vessel may have looked like are also utilized. The generic models need not to be identical to the original vessel, but are within a geometric transformation of it in most of its parts. The markings on the 3D surfaces of fragments and generic models are extracted based on their color cues. Ceramic fragments are then aligned against the corresponding generic models based on the geometric relation between the extracted markings. The alignments yield sub-scanner resolution fitting errors.

1. INTRODUCTION

1.1 Problems and Motivation in Pedestrian Tracking

Surveillance cameras are prevalently installed everywhere, but the videos are usually used only “after the fact” as a forensic tool thus losing its benefit as an active and real time media [1]. Tracking objects automatically can detect the abnormal behavior and suspicious individuals in real time and make preventing crimes possible. By recording large sets of tracks of people, designers can place nursing supplies at well-chosen locations in hospitals and place fountains and benches in public squares appropriately to make it more accessible to people [2]. Object tracking also plays an important part in many other computer vision applications such as crowd management, motion analysis and human-machine interaction.

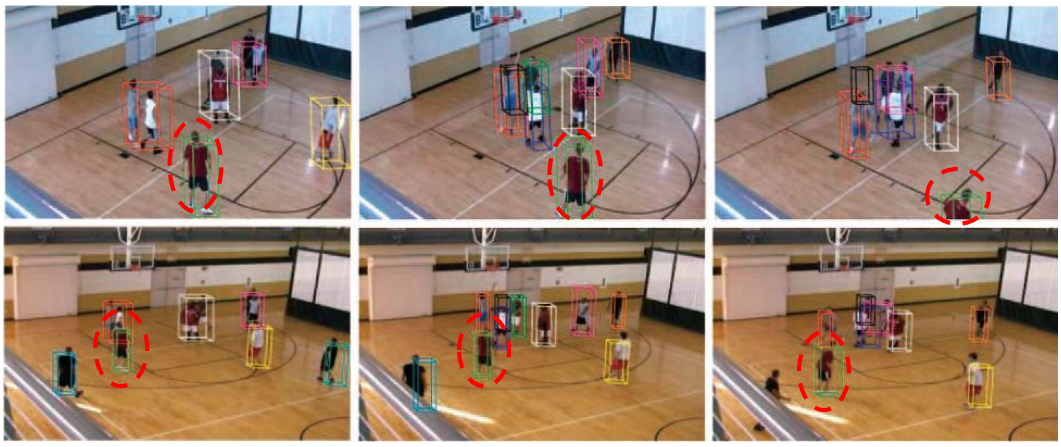


Figure 1.1: Tracking a basketball player by consistently labelling him across frames

Generally, object tracking can be divided into two categories. The first one is consistently labeling an object across time and cameras (if more than one camera is used). This can highlight the object of interest effectively. The first and second row in Figure 1.1 are frames captured by cameras 1 and 2. The man as shown in the red circle is tracked by being consistently labelled using green cuboid across frames and the two cameras. By tracking the basketball player, the coach can evaluate his performance and the player can improve himself accordingly.

With only one cameras, [3-5] track objects using global color reference models. The current frame is searched for a region, a fixed-shape variable-size window, whose color content best matches the color model of an object in the previous frame. This region is given the same label as the object in the previous frame. Instead of using the deterministic search in [3-5], Pérez et al. [6] introduce a new Monte Carlo tracking within a probabilistic framework. It can better handle color clutter in the background and complete occlusion over a few frames. Multiple cameras are used to label objects over time to resolve the occlusion problem that is apt to happen when using one camera. Chang and Gong [7] use Bayesian networks to combine the geometry-based modalities and recognition-based modalities for matching/labelling subjects between consecutive image frames and between multiple camera views. In [8], *N* points belonging to the medial axis of the upper body are used as the feature for tracking and multivariate Gaussian models are applied to find the most likely matches of human subjects between consecutive frames taken by several cameras. Khan and Shah [9] first assume that the single-camera tracking/labelling result is available, and then use the FOV (field of view)

lines to disambiguate multiple possibilities for correspondence between different cameras.

The second kind of tracking is to detect the objects' 2D/3D positions in real world or a reference coordinate system across time. This usually needs multiple cameras. 2D/3D trajectories are generated by associating the positions of an object across time. The tracking we perform in this thesis falls into this category. Figure 1.2 shows the 2D trajectories of the basketball players in a short period of time [10]. Based on the trajectories, coaches can do the motion analysis on opposing players and make offensive and defensive tactics accordingly.

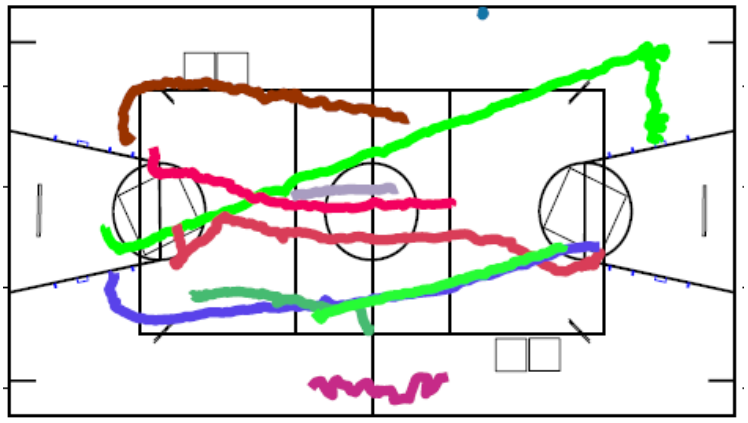


Figure 1.2: Ground plane trajectories of the basketball players

In [10], images from each camera view are projected on a top-view through multi-level homographic transformations to generate a detection volume for each object. Ground plane tracks are obtained using a track-before-detect particle filter that uses mean-shift clustering. Lee et al. [11] apply planar geometric constraints

to moving objects tracked throughout the scene. By robustly matching and fitting tracked objects to a planar model, they align the scene's ground plane across multiple views and decompose the planar alignment matrix to recover the 3D relative camera and ground plane positions. Mittal and Davis [12] propose a method which matches object regions along epipolar lines in each camera pair to obtain ground points guaranteed to lie inside the object. The estimates of 2D locations of objects are obtained by integrating results from camera pairs using outlier-rejection scheme. Those locations are then used to track objects over time. Most of the existing approaches for the tracking in the second category focus on generating the 2D ground trajectories and side view cameras are usually used. However, tracking people in 3D is also useful in many applications. For example, it can guide pan-tilt-zoom (PTZ) cameras to capture close-up face or iris images and analyze human behaviors such as sitting or falling down. In addition, tracking using overhead cameras have their own advantages over side view cameras and are rarely studied. In this thesis, we use overhead cameras to track people in 3D and use the 3D tracker to guide a set of PTZ cameras to capture high-quality close-up face images for recognition.

1.2 Problems and Motivation in Virtual Reconstruction of Broken Vessels

The archaeological journey from primary evidence collecting to public history interpretation has been long and arduous as analysis and meaningful history understandings are dependent upon time-consuming artifact reconstructions. To recreate an entire object from broken fragments, archaeologists need to manually

inspect the shape, color, material, texture, boundaries and fracture surfaces of fragments to join the fragments with adhesives and/or additional structural support materials, as shown in Figure 1.3. Often the mended pieces are kept in depositories and can't be shared with other labs as they are delicate and need to be well preserved.

Figure 1.3: Mending ceramic fragments manually

Reconstructing unearthed archaeological pieces virtually in 3D is motivated by the necessity to discover, preserve, and interpret history more efficiently. Many ceramic artifacts researched in the thesis are from one of the best preserved and most diverse urban colonial archaeological sites ever excavated - the Mall at Independence National Historical Park (INHP) in Philadelphia, Pennsylvania. This research on the whole is seen by INHP archaeologists as having a great potential to have significant implications for archaeological artifact mending, collections management, and site interpretation [13]. Once operated on a full scale, this technology will allow for more efficient laboratory work and will produce a significant time, money and labor savings. Computers (not just people) will be able

to mend the ceramic fragments based on the decorative markings (color patterns different from the background) on them to obtain 'piece back together' broken vessels. Such vessel reconstruction is a vital first step in the laboratory processing of artifacts. Speeding up this phase means faster advancement to the analysis and interpretation phase of study (as artifact identification precedes site analysis). Computer-assisted vessel reconstructions will furthermore allow for remote research capabilities as a collection of ceramics will be able to be studied off-site via digital proxies. Moreover, digital images and CAD visualization of reconstructed vessels will be a useful resource easily shared and be accessible all over the world through the Internet and will serve as an excellent educational tool. Records from this research and development project will be archived as a part of the INHP archaeological record collection in the Independence Park Archives.

Reconstructing a vessel virtually from broken fragments is similar to a 3D puzzle-solving problem. Intuitively, the information of the fracture surface of fragments such as the surface feature clusters [14] and surface normal [15] is used to find the matching pairs. But this doesn’t work well for thin-shell vessels since the information on the fracture surface is not enough. In this case, the contours of fractures are extracted and the matching fragments are established based on the curve matching [16-19]. However, all these methods are based on the fracture information of the fragments. As archaeological fragments’ edges may be eroded through time while in ground or during excavation, fracture information may not be well preserved as well, which means that neighboring fragments cannot be matched. In the thesis, we address this problem by utilizing surface information of

fragments (surface markings), which might be better preserved, in effectively represent and reconstruct fragments. We also integrate the experts' opinion by using generic models that are created by them. In this case, the vessels are virtually reassembled by aligning fragments to the corresponding generic model.

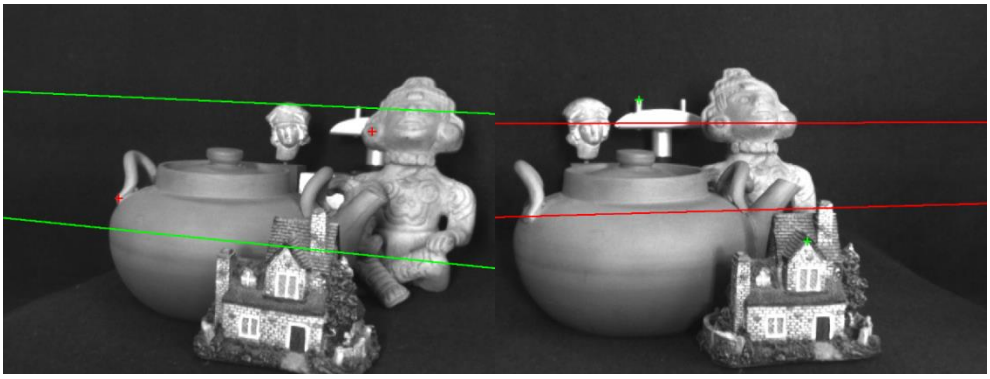
1.3 Geometric and Color Cues

Geometric and color information are two important aspects of computer vision. Geometric information such as the shape of an object, the geometric relations between multiple views of the same object or scene, the geometric properties of an object, is widely used in many applications. Objects can be detected and recognized based on their shapes [20], as shown in Figure 1.4(a) where three traffic signs are detected. Epipolar lines estimated using the geometric relations between the two views in Figure 1.4(b) help in finding corresponding points in the two views efficiently and accurately [21]. The corresponding points of the red points in the left image can be found on the red segments in the right image. The matching points of the green points in the right image can be detected in the same way. By using the geometric properties of the corner points of the chess board, the image in Figure 1.4(c) is rectified.

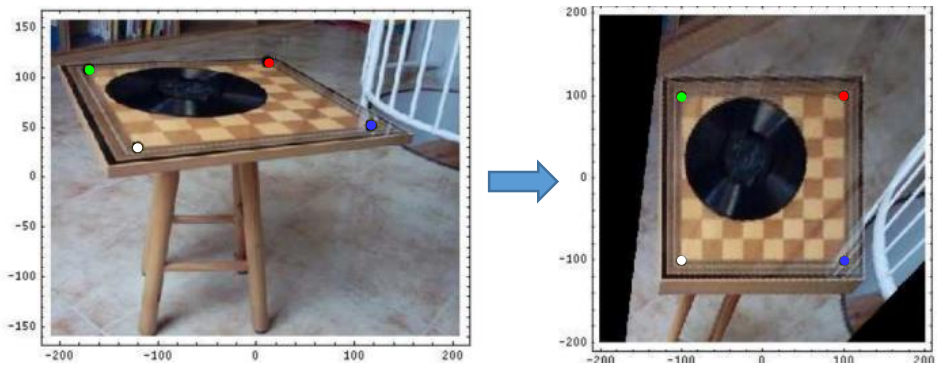
Color information such as color distribution, color histogram, color differences, is also very useful in a variety of applications such as in image segmentation [22], object recognition [23, 24], image enhancement [25], edge detection [26]. The image in Figure 1.5(a) is segmented into two parts based on spatially varying color distributions. Since color histograms are stable object representation in the presence

of occlusion and over change in view [24], based on which the four cups in Figure 1.5(b) are recognized as the same object. By adjusting the color histogram, the left image in Figure 1.5(c) is enhanced and more details are unveiled.

(a) Object detection



(b) Looking for corresponding points in 2 views



(c) Image rectification

Figure 1.4: Some applications based on geometric cues

(a) Image segmentation (b) Object recognition

(c) Image enhancement

Figure 1.5: Some applications based on color cues

In this thesis, both the geometric and color cues of the pedestrians/scenes are used to track pedestrians in crowded scenes. A 3D head point, the highest point of a person, is first obtained mainly based on the projective geometry. If the detected head points are not accurate, the color information across frames is utilized to detect and track them. The 3D head points can help to capture close-up face images. The two cues used in the 2D images are also utilized for 3D surface alignment to virtually reconstruct broken ceramic vessels. The markings are extracted from the

3D ceramic fragments based on the color cues, and then the fragments are aligned to the corresponding generic models using the geometric relations between them.

1.4 Contributions of the Thesis

The major contributions of this thesis are as follows.

In tracking:

1. It effectively deals with occlusion in crowded scenes by using two overhead cameras for tracking and smart PTZ cameras surrounding the scene for obtaining close-up frontal face images of a person of interest.

2. It achieves a very good approximation to an ideal tracker, which basically acts like a camera stuck and pointing to the person's face yielding 'good quality' face pictures for recognition and knows the 3D position of that person at all times.

3. It obtains salient and easily attainable points (head points) for 3D tracking using the projective geometry, and computes disparities only on the potential head top segment which is a short segment passing through the head top as opposed to everywhere in the scene.

4. It tracks people in 3D by fusing both the geometric and color cues. When the head points detected based on the projective geometry is not accurate enough, it combines mutual information between frames by using the color histograms associated with the head top area and its projection in the subsequent frames to better localize the head position, a requirement for acquiring high quality close-up face images of the person of interest at all time. The estimated velocity of a target

up to frame (*n-1*) is used to predict the search area where the head point at frame *n* resides.

5. It acquires close-up frontal face images for a person of interest through the use of distributed smart PTZ camera system, where the capture probability and camera handoff success probability (if a handoff between 2 cameras is needed) are quantitated and translated in terms of constraints on camera movement(s) and on its (their) physical parameters.

In vessel reconstruction:

1. Experts' knowledge and feedback are integrated in virtual reconstruction of broken vessels by using the generic models made by them.

2. It allows for built-in uncertainty of an expert model which is learned through approximations to the excavated vessel and/or through knowledge of the historical period.

3. Geometric and color cues of the ceramic fragments and generic models are combined to reassemble broken vessels. The markings on the fragment and generic models are extracted using the colors, and then the fragment is aligned against the corresponding generic models based on the geometric relations between them.

4. A novel set of affine weighted moments and absolute invariants are proposed to find the corresponding generic models and perform the alignment.

1.5 Organization of the Thesis

The rest of the thesis is organized as follows. In chapter 2, a novel approach is proposed to track pedestrians in 3D based on the head point detection using

overhead cameras. The pedestrian tracking is performed under both uncrowded and crowded scenes from an overhead view. For uncrowded scenes, mainly the geometric cues of the pedestrians are utilized, whereas the geometric and color cues are fused to detect and track pedestrians in crowded scenes. The tracking results in both scenes are given in the form of trajectories and the 3D tracking errors are provided. With the 3D tracking results generated by the fixed overhead cameras, we capture high-quality close-up frontal face images of a person of interest using a set of PTZ cameras across time in chapter 3. A PTZ camera scheduling scheme is presented based on the face image capture quality and PTZ camera handoff success probability both of which are quantified using mathematic models. In chapter 4, we propose to virtually reconstruct broken vessels by combining the geometric and color information of ceramic fragments. Markings on the fragments are extracted based on colors and then aligned to the generic models (which are created by the experts) using the geometric relations between them. Chapter 5 concludes the thesis and discusses about the possible extensions for the future work.

2. TRACKING PEDESTRIANS IN 3D

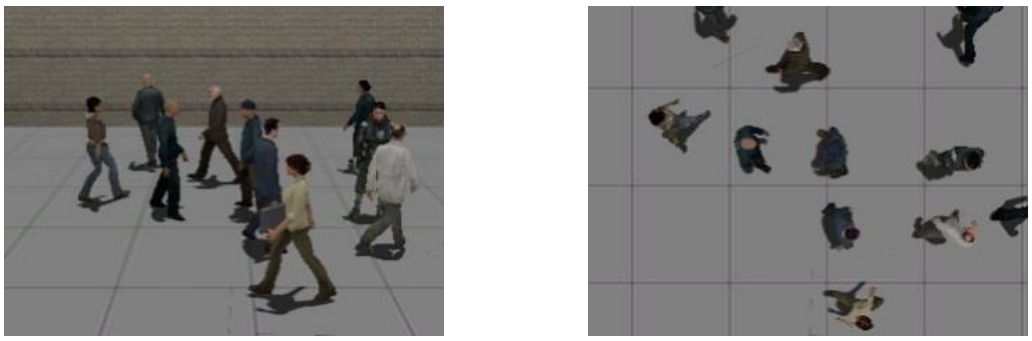
In this chapter, we propose to track pedestrians in 3D based on the head points using two horizontally aligned overhead cameras. Pedestrians are detected and tracked based on their geometric and color cues in the scene. A short segment passing through the head top of each person is detected using the projective geometry. A head point is detected on the segment and its 3D position is computed. The 3D head position is then tracked assuming constant moving velocity within two consecutive frames. For crowded scenes, the tracking is performed by using both projective geometry of the people and the color histograms of the head tops across frames. Our method works well for 3D tracking without using the full disparity map of a scene and improves the tracking robustness and accuracy by combining two complementary cues of the tracked persons. The experiments show that the average errors of the estimated ground plane positions and heights of the pedestrians in both uncrowded and crowded scenes are around 4 and 3 cm, respectively, and that our method is well suited for capturing close-up face images.

2.1 Introduction

With the prevalence of video surveillance, pedestrian tracking is drawing more attention and is more rigorously pursued. As the part of human body that least subjected to occlusion from different points of view, the human head is used to track people in a scene. Accurately tracking the 3D head position of a person is fundamental in capturing close-up facial images for most face recognition systems[27] and close-up iris images for iris recognition [28]. It is also helpful in

action recognition, such as fall detection of a person [29]. In this chapter, we focus on 3D pedestrian tracking based on head point in indoor environments, such as train stations, airports, shopping malls and hotel lobbies where scenes can be crowded.

Many existing tracking methods use a single side view camera [30-32], which cannot handle occlusions between people well, a phenomenon bound to happen in crowded scenes. To resolve the occlusion problem, multiple side view cameras are used. The more cameras are used, the more accurate the targets are localized. This, in turn, increases the computation and data transmission load as each view of the scene should be transmitted to the computer and processed.



(a) Side view (b) Overhead view

Figure 2.1: Occlusions under a side and overhead view of the same scene

Overhead cameras, which are usually deployed in indoor environments, offer advantages over side view cameras. As shown in Figure 2.1, occlusion is much less likely to happen in an overhead view compared to a side view where almost no person is viewed by him/herself. However, when the scene becomes more crowded, some people can be very close to each other and inter-object occlusion can occur.

In Figure 2.2, the pedestrians inside the circles are occluded and their images are connected. But their heads are always visible, thus tracking the head tops is very feasible to track people in 3D. In this chapter, we use two identical cameras that look straight down and are installed at the same height to track the pedestrians in uncrowded and crowded scenes, as shown in Figure 2.1(b) and Figure 2.2, respectively. Here the crowdedness is defined from an overhead view, and even an uncrowded scene from an overhead view where images of each person are separated from each other (as shown in Figure 2.1(b)) is crowded from a side view (as shown in Figure 2.1(a)).

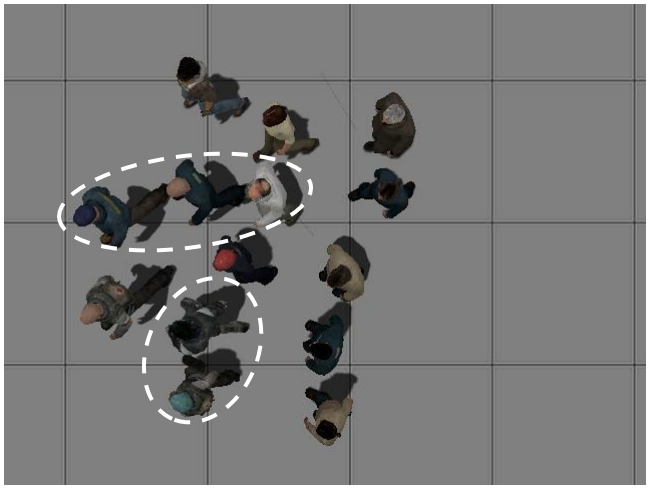


Figure 2.2: A crowded scene from an overhead view

We assume that people in a scene are upright and the head tops are their highest parts. This implies that the head top of a person is generally visible from an overhead view even in crowded scenes. A 3D head point is defined as the highest point of a person and is roughly the center of the head top from an overhead view.

A human body is roughly symmetrically distributed around an axis, the central vertical axis. The vertical axis intersects a person at the ground point at the bottom and the head point at the top. The crowded scenes mentioned in this chapter are not extremely crowded (e.g., subway station during rush hours in Tokyo), i.e., people's bodies are not touching each other even when they are in very close proximity.

The major contributions of this chapter are as follow: 1) using an image from just one camera, a potential head top segment is determined based on the geometric cue of pedestrians for each pedestrian; 2) combining the image from the other camera, the 3D head point is localized efficiently without using depth images; 3) detecting and tracking pedestrians by fusing the geometric and color cues of pedestrians(scenes). Our approach has the following advantages: 1) lower computation and data transmission load when compared to using several side view cameras; 2) no full disparity map of the scene is needed unlike other methods using stereo vision, resulting in a large saving on the computational load; 3) better scalability since the common FOV of two overhead cameras is rectangular and easy to be calculated and measured; 4) more accurate and robust tracking results by fusing two complementary cues.

2.2 Related Work

Side view cameras are extensively used to track people due to their applicability in both indoor and outdoor environments. Many researchers utilize a single side view camera to do tracking. Zhao and Nevatia [30] use ellipsoid human shape model to aid in foreground segmentation and resolving occlusions. Each

person is then tracked in 3D using a Kalman filter approach. Prince et al. [31] present a robust method for locating potential head regions using 3 weak cues: skin color, motion detection and foreground extraction to form three bottom-up likelihood maps. The probability of a face appearing at each image location is then yielded by combining these likelihood maps with spatial priors. Brostow and Cipolla [32] propose an unsupervised Bayesian framework for clustering simple image features to track persons in a crowd. By and large fully or partially occluded objects present a challenge to these methods.

To solve the occlusion problems, multiple side view cameras are deployed. Orwell et al. [33] propose to track objects in multiple views using color tracking. The connected blobs obtained from background subtraction are modelled using color histograms and are then used to match and track objects. Krumm et al. [34] combine information from multiple stereo cameras to detect human-shaped blobs in 3D space. Color histograms are created for each person and are used to identify and track people. Mittal and Larry [35] match object region based on the color characteristics in each camera pair. For each pair of matched region the back projection in 3D space is done in a manner that yields 3D points that guaranteed to be inside the object. Although these methods attempt to solve the occlusion problem, they may fall short due to either near total occlusion or when people are dressed in similar colors. To handle this problem, instead of using color or shape cues of a person, Khan and Shah [36] use a planar homography constraint that combines foreground likelihood information from different views to resolve occlusions and to determine regions on scene planes that are occupied by people. The homograghy

constraint, together with the multi-view geometric constraint regarding perpendicular projection of a camera's optical center is also used by Sun et al. [37]. Similar to our approach (either for uncrowded [38] or crowded scenes [39] from an overhead view), Eshel and Moses [40] focus on tracking people's head. They derive homograghy matrices at different height from the ground plane to align frames from different cameras and detect 2D patches using intensity correlation at various heights. The highest patch is regarded as the head patch. However, the thresholds of intensity correlation are set manually for each sequence and the method doesn't work on non-planar ground - an assumption needed for the use of the homograghy constraint. Multiple planes parallel to the ground are used in [41] to increase localization robustness. Similar to [42] and [43], our tracking method for crowded scenes [39] integrates the information of all parallel planes by considering all foreground pixels along a vertical segment.

For indoor environments, overhead cameras are also used. In [44] and [45] one overhead camera is used to localize a person, and the centroid of the foreground blob is taken as the ground position. The method is not accurate especially when people are close to the camera or walking around the boundaries of the FOV, and it fails when more than one person exists in a foreground blob. Boltes et al. [46] detect people's heads from an overhead view by placing pasteboards with markers on the heads. To reduce the perspective distortion error, the height of a person is needed and color coded as a marker on the pasteboard. This method, however, is not applicable in general scenarios because the assumption of known heights and requiring people to have markers on their heads are not practical. To obtain the 3D

position of a person without these constraints, stereo overhead cameras are applied and robust background subtraction is done in 3D space. Beymer [47] reprojects 3D points to a top-down orthographic view to track the people's ground position. Oosterhout et al. [48] detect 3D head positions in highly crowded situations by matching a sphere crust template on the foreground regions of the depth map and then track the head using Kalman filters. In another paper [49], they localize people in the scene by maximizing the similarity between the depth map obtained from a stereo camera and that reconstructed by projecting a certain number of templates at certain locations, as shown in Figure 2.3. By using stereo images, Boltes and Seyfried [50] build the perspective height field of pedestrians which are represented by a pyramid of ellipses, as shown in Figure 2.4. A person is then tracked using the center of the second ellipse from the head downward.

Figure 2.3: 3D template matching

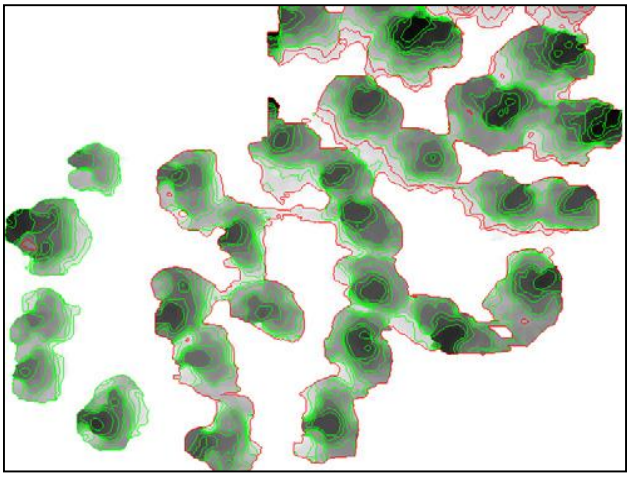


Figure 2.4: Perspective height field

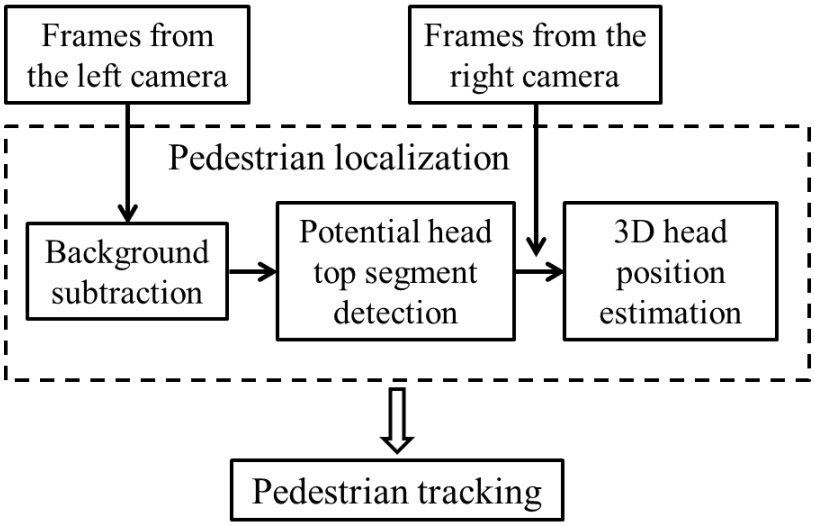


Figure 2.5: Flowchart of pedestrian tracking under uncrowded scenes

2.3 3D Pedestrian Tracking in Uncrowded Scenes

Figure 2.5 shows the overview of tracking in uncrowded scenes. First, the background subtraction is performed for frames from the left camera. For each

extracted foreground blob, which corresponds to a pedestrian in an uncrowded scene, a short segment passing through the head top, i.e., a potential head top segment is determined based on the projective geometric cues of the pedestrian. The head point of each person is detected on the segment by using the synchronized left and right images and the 3D position is calculated using triangulation. Once being localized, pedestrians are tracked across frames based on the constant moving velocity within two successive frames.

2.3.1 Background Subtraction

Every tracking method requires an object detection mechanism either in every frame or when the object first appears in the video. Typically, the common approach for detecting objects from a background scene is background subtraction: building a background model and then finding deviations from the model for every frame. A color balancing of background image and the frame based on the gray-world assumption is taken against global illumination changes before background subtraction by Oto et al. [51]. Wren et al. [52] model the color of each background pixel with a 3D (Y, U and V color space) Gaussian. The mean and covariance are learned from the color observation in several consecutive frames, thus can cope with illumination changes. But Gao et al. [53] point that a single Gaussian is not a good model for outdoor scenes since multiple colors can be observed at a certain location due to repetitive object motion, shadows, or reflectance. To cope with this problem, Stauffer and Grimson [54] use a mixture of Gaussians to model the pixel color. While Horprasert et al. [55] propose a color model (brightness distortion and chromaticity distortion) which separates the brightness from the chromaticity

component. This method helps to distinguish shading background from the ordinary background and moving foreground objects, thus is able to handle the local and global illumination changes.

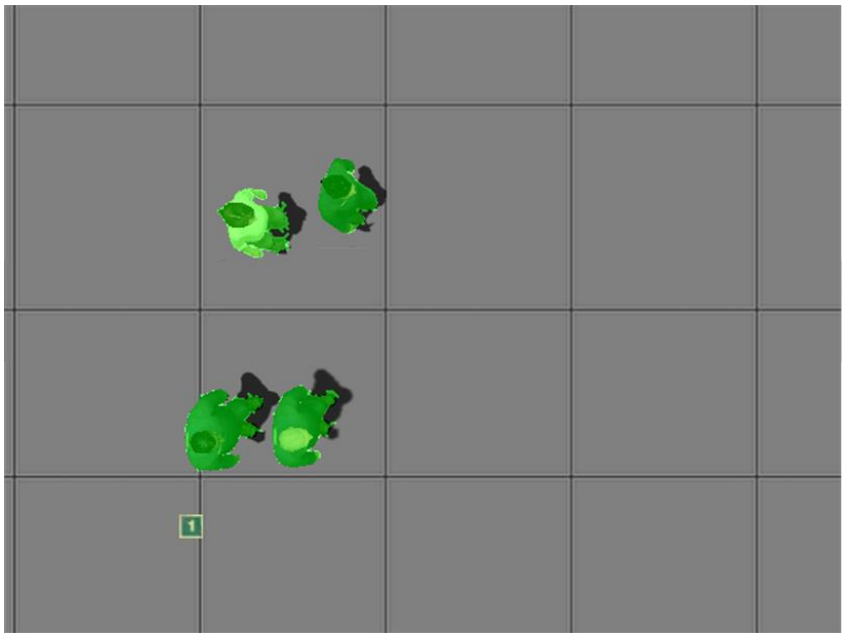


Figure 2.6: Extracted foreground blobs in a frame from the left camera

In this thesis, the foreground blobs of pedestrians are extracted using background subtraction done in hue-saturation-value (HSV) color space, which separate the brightness from the hue and saturation components, rather than in the RGB color space to remove the shadow caused by the lighting. This shadow removal technique is sufficient for indoor environment, such as train stations, airports and hotel lobbies, where the shadows are small and diffused. After background subtraction, the small ‘holes’ inside the foreground area are removed using binary area openings [56]. The extracted foreground areas are shown as the

green regions in Figure 2.6. We can see that the shadows are excluded from the extracted foreground. Unlike the method using multiple side view cameras, the foreground segmentation is only implemented for frames captured from one overhead camera (the left one in this thesis), which makes our approach more efficient.

2.3.2 Potential Head Top Segment Detection

In this section, a potential head top segment that contains head top points is detected for each extracted blob in the image from the left camera.

Figure 2.7 shows the geometric relationship when an image of a person is taken by an overhead camera *S*. A person is simplified as a cylinder model with the black dashed line as the central vertical axis *l*. *G* is the ground point where a person touches the ground. π is the plane perpendicularly intersecting the central vertical axis of the person at the ground point *G*. If the ground is flat, π is the ground plane since the person is walking upright. Otherwise π is a hypothetical plane. *O* is the perpendicular projection point of the optical axis of the overhead camera on plane π. Since both the optical axis of the camera and the central vertical axis *l* of the person are perpendicular to the plane π, they are in the same plane, plane *SOG*. The perspective projection of central vertical axis *l* lies on the line $\overleftrightarrow{OG}$. Thus with the assumption in section 2.1, it can be inferred that the highest head top point lies on the line $\overleftrightarrow{OG}$ as well; and the shadow area *A*, the projected area of the person on plane π from the camera's view, is divided into two halves by the line $\overleftrightarrow{OG}$, the projection of the person's symmetrical axis. The point *C* as the centroid of the area

A should be also on the line $\overleftrightarrow{OG}$ which is denoted as $\overleftrightarrow{OC}$ for later use. F is the furthest point defined as

$$F = \underset{F \in A, F \in \overleftrightarrow{OC}}{\operatorname{argmax}} |\overline{OF}| \tag{2.1}$$

From the assumptions in section 2.1, we can safely argue that the segment $\overline{CF}$ passes through the projected head top on plane π no matter the ground is flat or not.

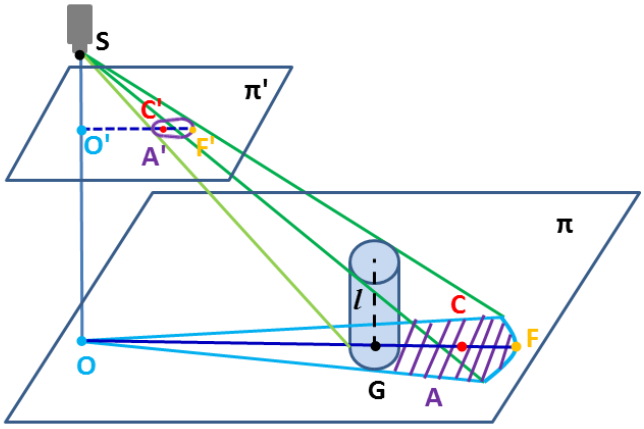


Figure 2.7: Detection of the potential head top segment using projective geometry

Plane π' shown in Figure 2.7 is the image plane of camera S, and A' is the image of A, i.e. the foreground blob of the person. F' is the furthest point on the image plane, similarly to equation (2.1), defined as

$$F' = \underset{F' \in A', F' \in \overleftrightarrow{O'C'}}{\operatorname{argmax}} \left|\overline{O'F'}\right| \tag{2.2}$$

where C', the image point of C, is the centroid of the pedestrian's foreground blob A' and O' the image center. So the head top pixels can be found on segment $\overline{C'F'}$. The furthest point on the image plane F' and the blob centroid C' are shown in

Figure 2.8 as the green and red point. For an upright pedestrian, the furthest point F' can be a pixel on the head top, as shown in Figure 2.8(a) or other parts of the body, mainly the shoulder as shown in Figure 2.8 (b), depending on the walking direction and his/her location relative to the FOV center. In either case $\overline{C'F'}$ always passes through the head top pixels.

(a) The Furthest point on the head top (b) The Furthest point on the shoulder

Figure 2.8: The centroid and the furthest point of a person

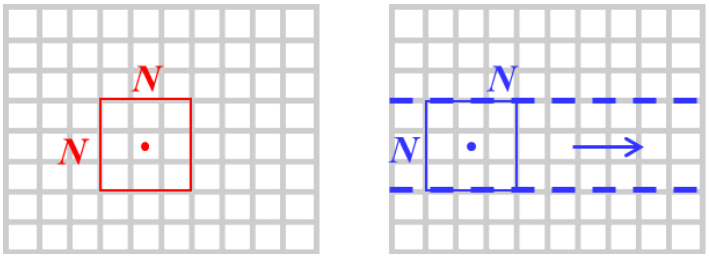
2.3.3 3D Head Position Estimation

The 3D head point is where the central vertical axis of a person intersects the head top, i.e. the highest point of the head top and the whole body. In section 2.3.2, a short segment containing head top points of a person is obtained from images captured from the left camera. Thus locating the 3D head point reduces to finding the center of points on the segment which are closest to the cameras and thus have the largest disparity.

2.3.3.1 Establishing Disparities along the Potential Head Top Segment

To calculate the disparity of each pixel on the potential head top segment, its corresponding pixel needs to be found on the synchronized right image. For each pixel on the segment from the left image, we compare the *N***N* region about this pixel (the template) with a series of regions of the same size extracted from the right image (the samples) based on the RGB value distributions, as shown in Figure 2.9. The center of each sample, the candidate matching pixel, has the same row number as the pixel in the left image, since the left and right camera are aligned horizontally. The search range on the row can be largely narrowed down with the disparity range of a head point. A pixel on the potential head top segment is described as an *N***N**3 vector $\boldsymbol{L}$, containing the RGB values of all pixels in the template. The k^{th} candidate matching pixel is described by the same size vector $\boldsymbol{R_k}$. The similarity of the two vectors is evaluated using Euclidean distance

$$D_k = ||\boldsymbol{L} - \boldsymbol{R}_k|| \tag{2.3}$$



(a)Left image: the template (b) Right image: samples

Figure 2.9: Finding the corresponding pixel from the right image (b) for a pixel on the potential head top segment in the left image (a)

A corresponding point of the point on the potential head top segment is established if

$$D_a < \gamma \cdot D_b \tag{2.4}$$

where D_a and D_b are the minimum and second minimum of D_k and γ is the similarity ratio (typically $\gamma = 0.8$). The disparity of the point on the segment is computed from the difference of the two matching pixels.

To get a more accurate 3D position, we estimate the sub-pixel disparity by considering D_a that satisfies (2.4) and its two neighboring values instead of just taking the point with minimum D_k as the matching point. A parabola is fitted to the three values and the minimum is analytically solved for to get the sub-pixel correction. The disparities on the potential head top segment may have outliers caused by mismatching of the pixels from the left and right image. The 3D head point detection will be highly affected if the outlier has the maximum disparity on the segment. A correct disparity distribution along the segment has only one peak because of the 'Ω' shape of the upper body. To remove the outlier robustly, the disparities are first rounded to integers. If there are multiple local maxima in the rounded disparity distribution, those looking like spikes are considered as the outliers and removed.

2.3.3.2 Computing 3D Head Coordinates

After outlier removal, the center of the pixels with the largest rounded disparity instead of only the pixel with largest disparity on the potential head top segment is determined as the head point on the image plane. When computing the 3D head location, the largest disparity (not rounded) is used. This way, both the disparity

and the image plane position can have sub-pixel resolution, making the localization of the 3D head point more robust and accurate.

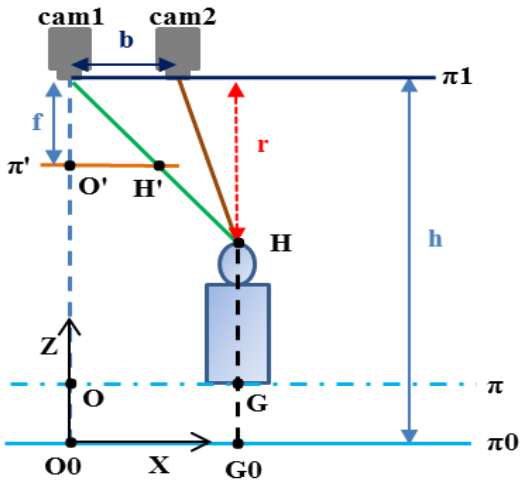


Figure 2.10: 3D head position calculation

Figure 2.10 is the front view of the scene with a person in it. *cam1* and *cam2* are the left and right overhead camera. The two cameras have a common image plane π' because of the way they are deployed. Plane $\pi 1$ is parallel to the image plane π' and contains the optical centers of *cam1* and *cam2*. *H* is the 3D head point which is the closest point of the person to plane $\pi 1$ with the distance

$$r = bf/d \tag{2.5}$$

where f is the camera focal length, b the baseline length and d the disparity of the 3D head point. The head pixel H' is the 3D head point in image plane π'.

In Figure 2.10, the reference plane $\pi 0$ is defined as the plane $z = 0$. The projection point of the optical centre of *cam1*, *O0*, is the origin of the world

coordinate system, and the x-axis is parallel to the baseline. The z-axis is the optical axis of *cam1*. The y-axis is determined by the right-hand rule. *G* is the ground point of the person and is the point where plane π perpendicularly intercepts the central vertical axis of the person. *G0* is the perpendicular projection of *G* on the reference plane. If the ground is flat, $\pi 0$ is the ground plane and is overlapped with π (*G* and *G0* are also overlapped). From the stereo triangulation, we can compute the 3D position of head point *H (x, y, z)* as

$$(x, y) = \overline{O'H'} * r/f, z = h - r \overset{(2.5)}{\Longrightarrow}$$
$$(x, y) = \overline{O'H'} * b/d, z = h - r \quad (2.6)$$

where $\overline{O'H'}$ is a vector in the image plane showing the position of head pixel H' relative to the image center O' and h is the camera height. f , b and d have the same definition as in equation (2.5). (x, y) in (2.6) is the coordinate of the head point in the image plane π'. To get the 3D location, (x, y) is converted to the position in the coordinate system of the reference plane $\pi 0$ using simple calibration of the overhead camera.

2.3.4 Pedestrian Tracking

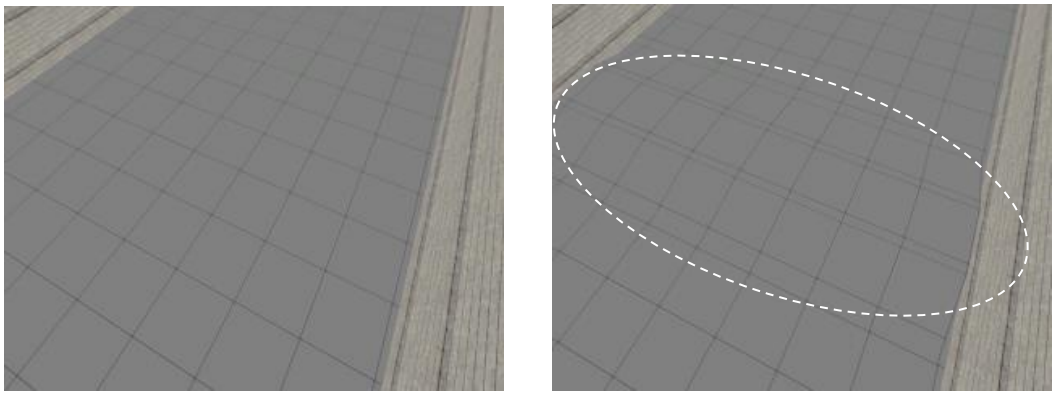
Once the 3D positions of pedestrians, denoted as the 3D head points, are obtained in each frame, they are tracked by assuming a constant moving direction and velocity within two consecutive frames. With the position of a person in the previous frame, his/her position in the current frame is predicted using the assumption and a search is implemented in a neighborhood around the predicted point. The position of the person is then updated by the detected 3D head point that is nearest to the predicted point and within the search area. If no head point is found

in the search area, the person's location is updated using the predicted one. The person is deleted if he or she is not found over certain extend period of time. Similarly, if an object is not associated with any object in the previous frame over some frame intervals, it is regarded as a new target.

2.3.5 Experiments

2.3.5.1 Experiment Setup

We test our approach using a publicly available visual surveillance simulation test bed, ObjectVideo Virtual Video (OVVV) [57]. OVVV is based on a commercial game engine, which can make human models in the scene behave like people in real world. It allows placing and configuring static and pan-tilt-zoom (PTZ) cameras freely and can generate the true 3D position of an object as well. Some researchers [58, 59] also test their algorithms on the similar virtual environments such as a virtual reconstruction of the original Penn Station in New York City [60].



(a) Planar ground (b) Non-planar ground

Figure 2.11: The virtual scenes of the train station concourse

Two virtual scenes of the train station concourse are created, one with flat ground (Figure 2.11(a)) and the other with a small bump, whose cross section is a trapezoid, added on the flat ground (the area inside the ellipse in Figure 2.11 (b)). Seven people are walking in an area of about 4*4.5 m, which is an uncrowded scene (however, it's crowded from side views): the blobs of people don't merge in the overhead view of a scene.

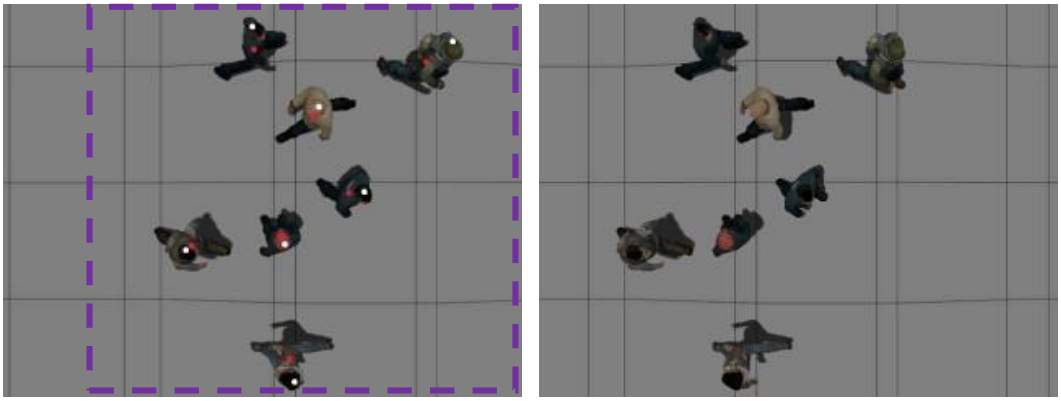
We set the flat ground and the flat part of the non-planar ground (i.e., the bottom of the bump) as the plane z=0. The ceiling is 8.84 m high from the plane *z*=0. Two identical synchronized cameras are installed on the ceiling with perpendicular views. The baseline is horizontal and the length is 1 m. The frame rate in both scenes is set as 15 frames per second in the test bed and the frame size is 640*480 pixels for overhead cameras.

2.3.5.2 Experiment Results

We compare our method with blob centroid (BC) based method which uses blob centroids as the ground plane positions [45]. Both approaches are applied to track people in the scene with planar and non-planar ground. Since the BC method only uses one camera, the left camera in our experiments, it can only estimate the ground (X-Y plane) positions and we only compare the ground tracking results. To use the BC method in the scene, the detected blob center of each person is assumed to be on the plane z=0 for both planar and non-planar ground. Its 2D position in real world is calculated by converting the coordinate system of the image plane to that of plane z=0 using simple calibration of the left camera.

(a) Left image (b) Right image

Figure 2.12: The frames captured by two cameras with people walking on the planar ground



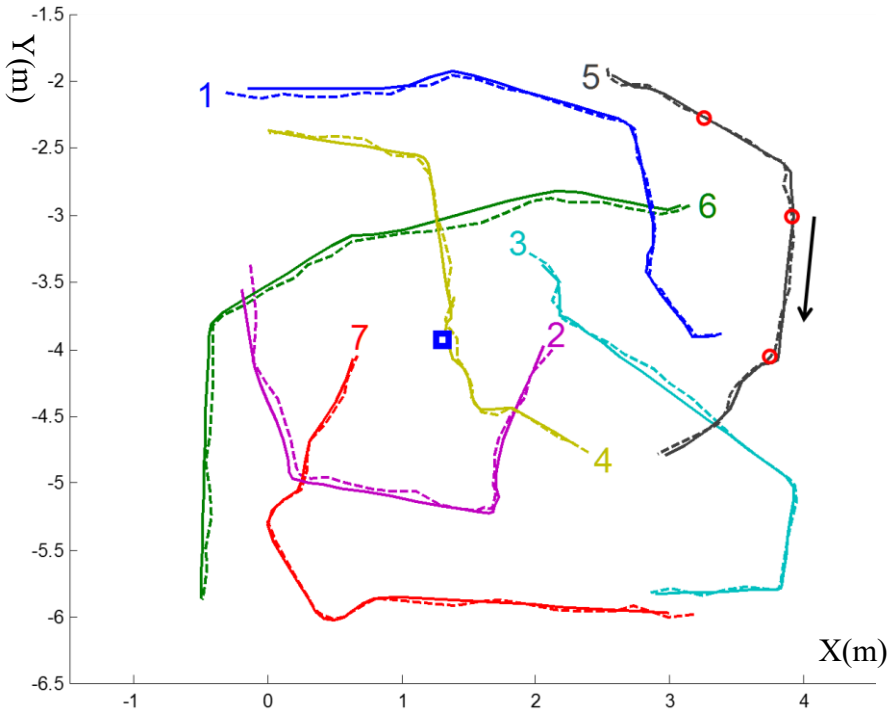
(a) Left image (b) Right image

Figure 2.13: The frames captured by two cameras with people walking on non-planar ground

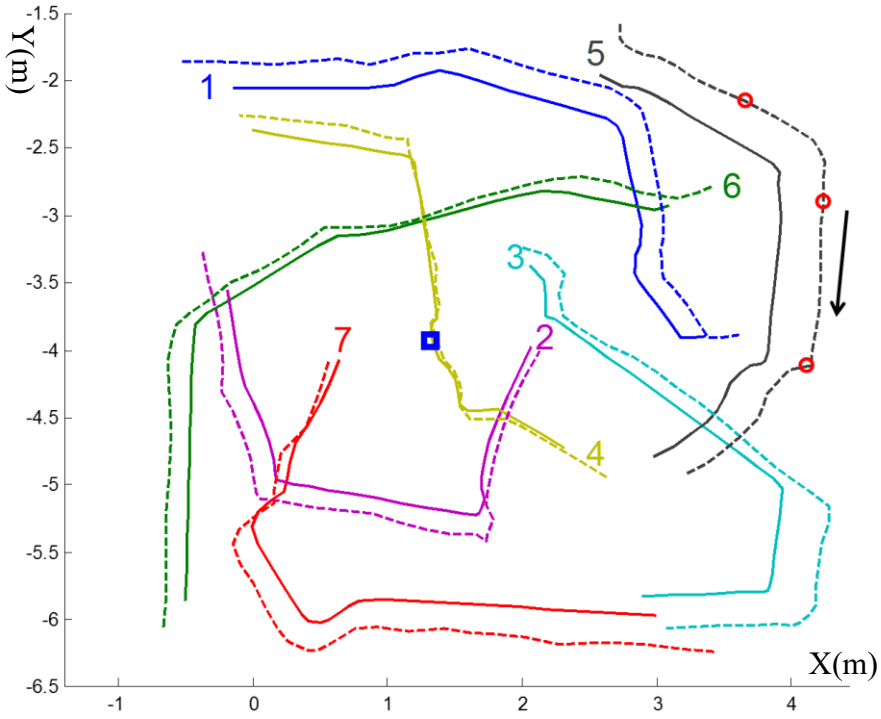
First we let a group of people walk on the planar ground and then let the same group of people walk on the non-planar ground using the same paths. Figure 2.12 and Figure 2.13 show the images captured by the two overhead cameras when people are walking on the planar and non-planar ground, respectively. The

foreground centroids and the detected head points are marked as in red and white in the left image. The detected head points are very close to the head top centers in both scenes, and are different from the blob centers, especially when a person is far away from the FOV center. The dashed line square in Figure 2.13 shows the bump area. The two sets of images in Figure 2.12 and Figure 2.13 are captured almost at the same time. But they don't look the same since the bump actually affects their walking velocities.

Our approach can estimate the 3D tracks for pedestrians. To display our results more clearly, the 3D tracks are projected to the X-Y plane and Z plane separately. This also helps to guide a PTZ camera, since the X-Y value and Z value are used to determine the pan angle and tilt angle, respectively. The X-Y plane tracking results using our approach and the BC method in the two aforementioned scenes are shown in Figure 2.14 and Figure 2.15, where the solid lines are the ground truth provided by the test bed and the dashed lines are the estimated trajectories. The small blue square is the FOV center of the left camera. The number at the one end of each trajectory denotes its object ID and the same pedestrian in both scenes is given the same ID. From Figure 2.14 and Figure 2.15, we can see that the X-Y plane trajectories obtained using our method is very close to the ground truth, while the method using the blob centroids can only track the pedestrians accurately when they are very close to the FOV center. The further a person is away from the FOV center, the larger the tracking errors are. Because of the bump that changes people's speeds according to which part of the bump they are at, the trajectories in the two scenes are a little different although the same paths are set.

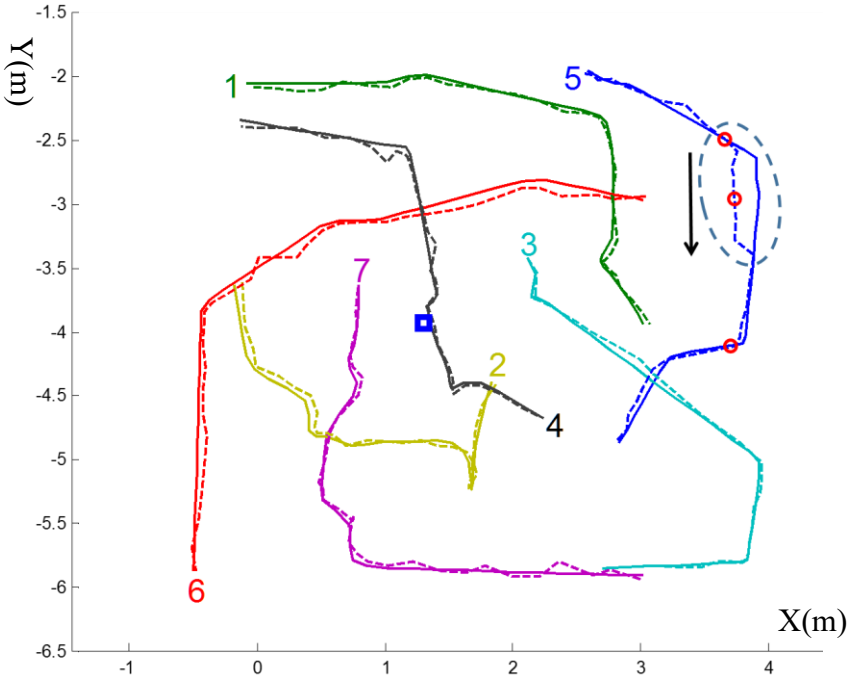


(a) Trajectories obtained using our approach

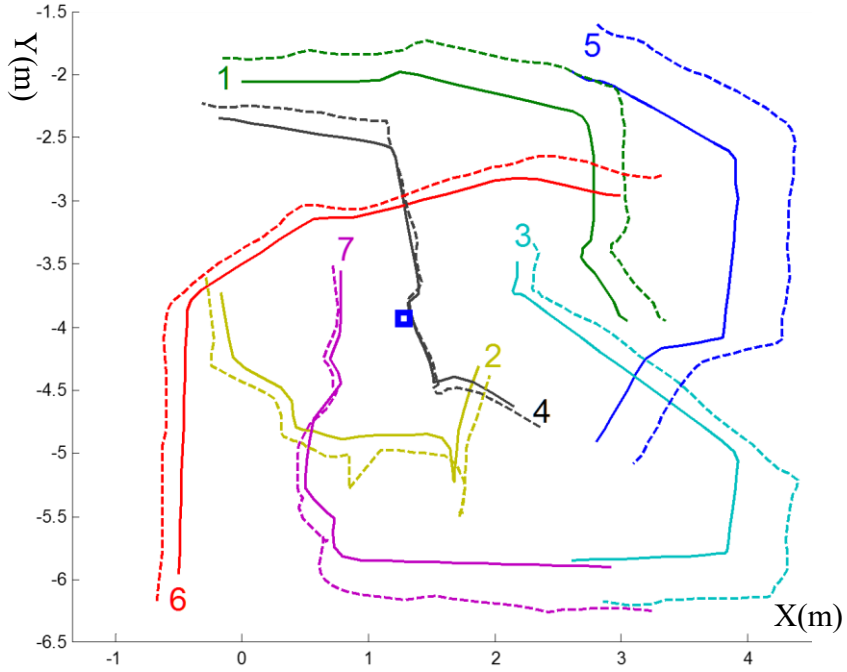


(b) Trajectories obtained using the BC method

Figure 2.14: X-Y plane tracking results using two different methods when people walking on the planar ground



(a) Trajectories obtained using our approach



(b) Trajectories obtained using the BC method

Figure 2.15: X-Y plane tracking results using two different methods when people walking on the non-planar ground

Table 2.1: The errors of the estimated tracks on the planar ground

Object ID	BC method X-Y errors(cm)	Our method	
		X-Y errors (cm)	Z errors (cm)
1	20.54±2.50	3.42 ± 2.21	2.87 ± 1.31
2	15.00±3.85	4.76 ± 2.07	3.76 ± 4.43
3	25.00±10.87	2.74 ± 1.75	3.14 ± 3.26
4	8.29±5.15	4.31 ± 3.00	4.32 ± 3.46
5	31.11±3.03	3.28 ± 1.44	2.58 ± 3.11
6	15.23±6.03	4.78 ± 2.22	2.62 ± 1.98
7	20.13±6.91	2.63 ± 1.79	3.14 ± 0.45

Table 2.2: The errors of the estimated tracks in the non-planar ground

Object ID	BC method X-Y errors(cm)	Our method	
		X-Y errors (cm)	Z errors (cm)
1	26.95±4.30	3.74 ± 1.89	2.90 ± 1.57
2	16.29±4.79	3.66 ± 1.86	3.63 ± 4.68
3	36.48±14.24	2.40 ± 1.82	3.08 ± 2.98
4	10.91±6.12	4.16 ± 2.63	3.71 ± 3.53
5	43.18±4.85	6.17 ± 6.22	5.22 ± 4.16
6	17.66±7.01	3.78 ± 2.22	3.90 ± 4.21
7	21.76±12.27	4.82 ± 2.54	3.02 ± 4.24

The errors of estimated Z values together with the X-Y plane values for each person in the two scenes are tabulated in Table 2.1 and Table 2.2. Table 2.3 shows the overall tracking errors for the two methods. From the tables, we can see that the X-Y plane tracking errors of our approach is much smaller than those of BC method

and the errors in Z plane is very small. The 3D head position errors result from the two main reasons: a) the estimated potential head top segment is off the head top center due to pedestrians' movement which makes the foreground blob not perfectly symmetrical about the projection of the vertical central axis; b) robust corresponding points are not found on the head top part of the segment or are not established correctly between the images from the two cameras.

Table 2.3: The overall tracking errors

	BC method X-Y errors(cm)	Our method	
		X-Y errors (cm)	Z errors (cm)
Planar plane	19.33±9.14	3.70±2.26	3.21 ± 2.89
Non-planar plane	24.75±13.63	4.10 ± 3.25	3.64 ± 3.78

In Figure 2.16(a), a potential head top segment (the white segment) misses the head top centroid but the head point, the green dot, is still detected on the head top. This results in an error on X-Y plane that is usually smaller than the head top radius (8 cm on average for adults), and the estimated Z value is very close to the true value since the head top is relatively flat. The reason b) can cause relatively big error on both X-Y and Z plane, but this rarely happens. The ellipse in Figure 2.15(a) highlights the relatively big difference between the true and estimated X-Y plane tracks of person 5 for whom the average X-Y and Z errors are greater than 5 cm (shown in Table 2.2). This is because no robust matching points on the head top are found between the left and right images in some frames. Figure 2.16(b) shows the

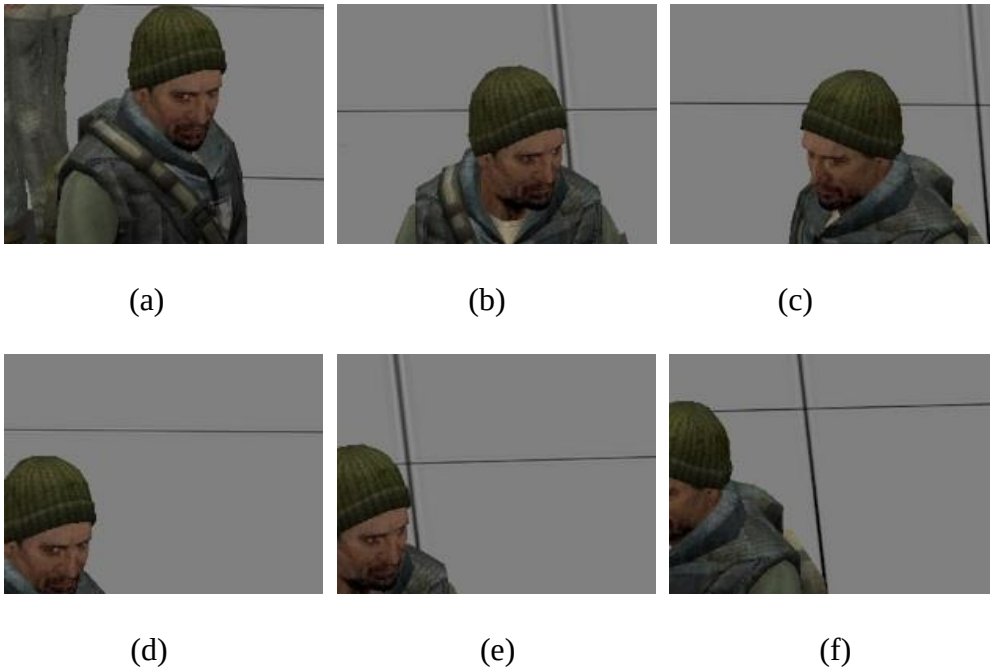
detected head point (the green dot) of person 5 in one of those frames. The head top potential segment is established relatively well but a point on the edge of the head top is detected as the head point, thus both the X-Y and Z errors are relatively big.

(a) (b)

Figure 2.16: A part of the frames showing (a) A not well detected potential head top segment and (b) A not well detected head point. The white segment denotes the potential head top segment, and the green and red dots denotes the foreground blob center and detected head point respectively

A PTZ camera is installed on the wall in both scenes with a resolution of 320*240 and a height of 4 m to capture the close-up facial images. The X-Y plane position of the PTZ camera is (3.81, -8.89) m with the same coordinate system as shown in Figure 2.14 and Figure 2.15. To compare the facial image capturing results based on the head points detected by our approach and the BC method, an additional PTZ camera is installed 25 cm above the existing PTZ camera in the scene with a flat ground to focus on the face of the same person. For the scene with non-planar ground, no additional PTZ camera is installed and the reason will be given in the next paragraph. The two PTZ cameras (the existing and additional one)

in the scene with planar ground and the one in the scene with non-planar ground are assigned to capture the close-up facial images of person 5 over time. The PTZ cameras are set to focus on the target with a small FOV of 1.27*0.95 m.



(a) (b) (c)

(d) (e) (f)

Figure 2.17: The close-up facial images captured in the scene with planar ground when using our method ((a)-(c)) and BC method ((d)-(e)), respectively.

(a) (b) (c)

Figure 2.18: The close-up facial images captured in the scene with non-planar ground by using the 3D head points detected by our approach.

Figure 2.17 shows the capturing result in the scene with planar ground, where the first row is the close-up facial images captured based on the 3D head location estimated by our method and the second row is obtained based on the BC method. Since Z values are not estimated in the BC method they are set as the average height of adults. The images in the first and second row from left to right are captured when person 5 arrives at the locations marked by the circles in Figure 2.14 (a) and (b) and the walking direction is shown as the black arrow. When the ground is non-planar, the Z value of a head point, i.e., the height from the head point to the plane z=0, cannot be obtained using the BC method even with the assumed average height, since the height of the bump where the person stands is unknown. Thus it's almost impossible to capture face images based on BC method and only 1 PTZ camera is installed in this case. However, with the 3D head points detected by our approach, the target's face images shown in Figure 2.18 are captured at the positions marked by the circles in Figure 2.15(a).

From Figure 2.17(a)-(c) and Figure 2.18, our method is very effective in capturing high quality close-up facial images, with almost all the captured faces around the image center. Even for the point inside the ellipse in Figure 2.15 (a) where both X-Y and Z plane errors are relatively large, the whole face is still captured although it's not in the very center as shown in Figure 2.18 (b). However, the BC method can't focus well on the face even the Z value (height) is assumed. To always capture the front view facial images, more PTZ cameras and a scheduling algorithm are needed, which will be introduced in the next chapter.

2.4 3D Pedestrian Tracking in Crowded Scenes

$\overline{C_1F_1}$ in Figure 2.19 is the potential head top segment obtained using the method in the section 2.3 for uncrowded scenes where the foreground blobs don't merge from an overhead view and a head point can be detected on $\overline{C_1F_1}$. However, in a crowded scene, the foreground blobs can be connected shown as these inside the circle in Figure 2.19. We can see that not all three head points lie on the potential head top segment $\overline{CF}$ detected using the same method. In this section, a novel approach is proposed to track people in crowded scenes with flat ground.

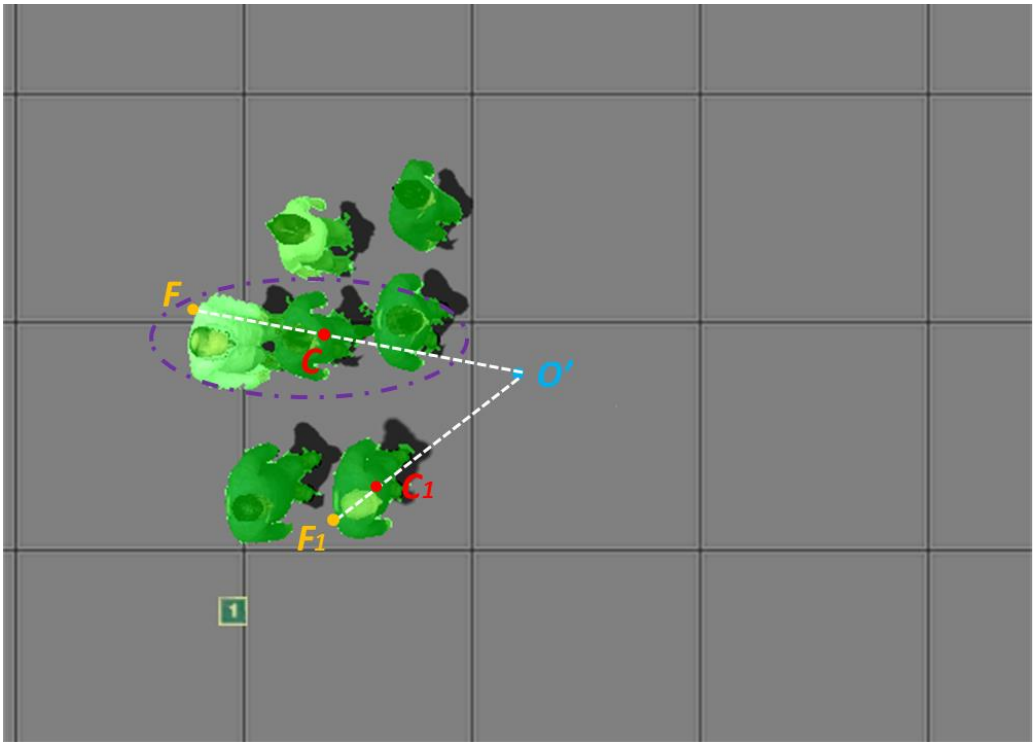


Figure 2.19: Potential head top segment detected using the method for uncrowded scenes

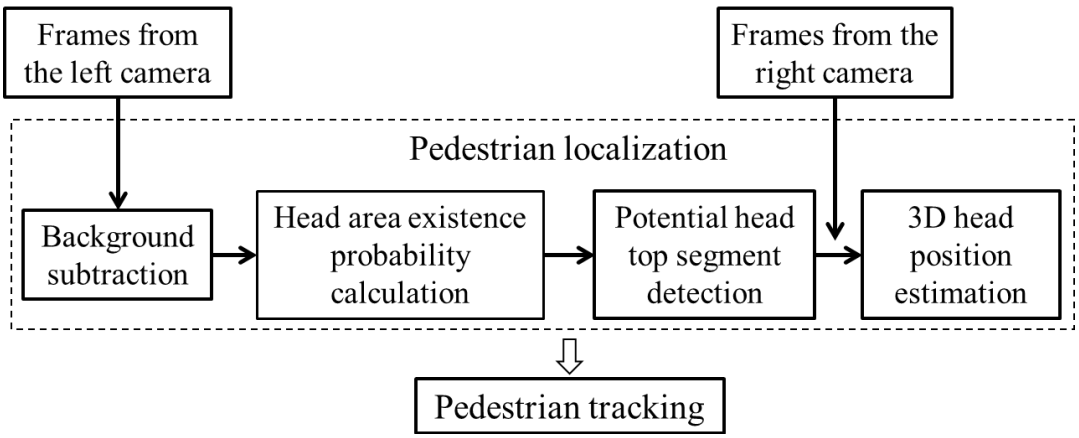


Figure 2.20: Flowchart of Pedestrian tracking in crowded scenes

The overview of our method is shown in Figure 2.20. First, the background subtraction is performed for frames from the left camera using the same method as in section 2.3.1. The existence probability of a head area is calculated for each foreground pixel based on the properties of the projective geometry of the pedestrians caught by the overhead camera. The regions consisting of the pixels whose existence probabilities are higher than a threshold are regions of interest (ROIs) and are clustered according to the distances between them if there is more than one ROI in a foreground blob. A short segment passing through the head top, i.e., the potential head top segment, is then established based on the centroid of the clustered ROIs. On this segment, the 3D head point is detected and its position is estimated using the method in section 2.3.3. People are then tracked across frames under the assumption of constant moving velocity (people walk in a non-erratic and smooth way). If the detected head point is not accurate enough, it is corrected and updated based on the similar color distribution of a head top within two consecutive frames. Since the background subtraction and 3D head position calculation in

crowded scenes use the same method as in section 2.3, we only present the calculation of head area existence probability, potential head top segment detection and pedestrian tracking in this section (i.e., section 2.4).

2.4.1 Head Area Existence Probability Calculation

To detect a potential head top segment, the head area existence probability for each foreground pixel is first calculated.

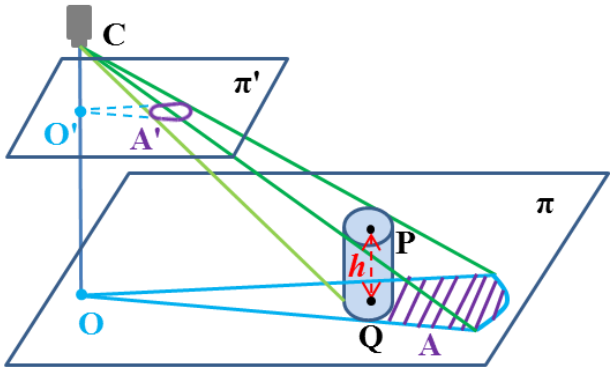


Figure 2.21: Projective geometry of an overhead camera

The probability of a foreground pixel being inside a head area is referred as the head area existence probability. We roughly model a person as a cylinder as illustrated in Figure 2.21. Point *P*, with the height *h*, is a point on a person and the cylinder whose diameter is the average human width extends from the ground plane π to the height *h*. The shadow area *A* in the plane π is the projection of the cylinder. A' in the image plane π', corresponding to *A*, is the image of the cylinder. Figure 2.22 shows a part of the image captured by the left camera, where O' is the image

center. From Figure 2.21, O' is also the vanishing point of all vertical segments on the image plane. A 3D points located on a vertical segment is projected on a vanishing line. In Figure 2.22, P' is the image of P and the segment along the vanishing line $\overline{P'Q'}$ denotes the height h in the image plane. Usually the longer $\overline{P'Q'}$ and the more number of foreground pixels (highlighted in green) inside A' indicate a larger probability that P' is within a head area.

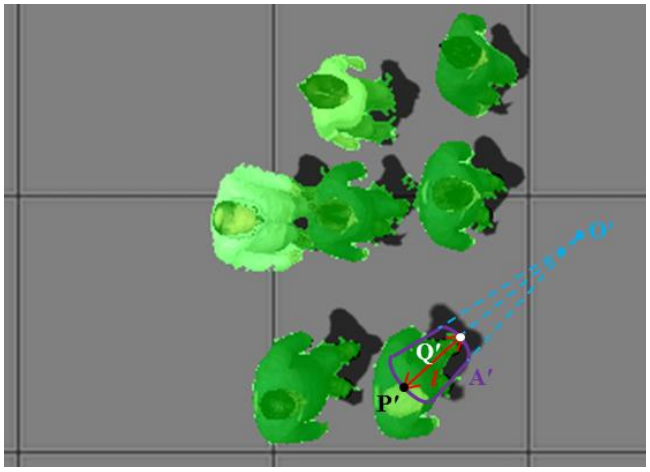


Figure 2.22: Part of the image captured by the left camera as shown in Figure 2.23(a)

To efficiently calculate both l (the length of $\overline{P'Q'}$) and the foreground area within A', a polar mapping is performed for each left image using the conversion formulas

$$\begin{cases} r = \sqrt{x^2 + y^2} \\ \theta = atan2(y, x) \end{cases} \tag{2.7}$$

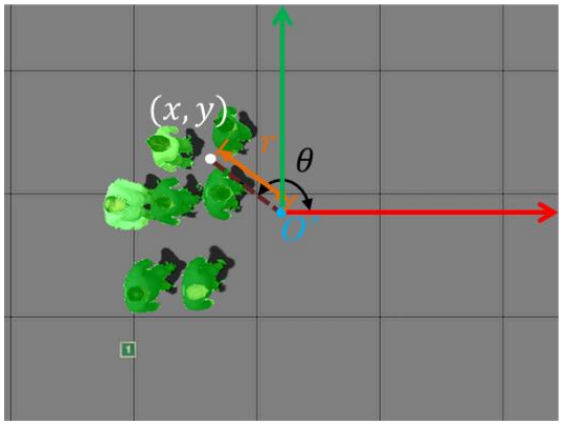
where $atan2(y, x)$ is defined as

$$atan2(y,x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & if\ x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & if\ x < 0\ and\ y \geq 0 \\ \arctan\left(\frac{y}{x}\right) - \pi & if\ x < 0\ and\ y < 0 \\ \frac{\pi}{2} & if\ x = 0\ and\ y > 0 \\ -\frac{\pi}{2} & if\ x = 0\ and\ y < 0 \\ 0 & if\ x = 0\ and\ y = 0 \end{cases} \quad (2.8)$$

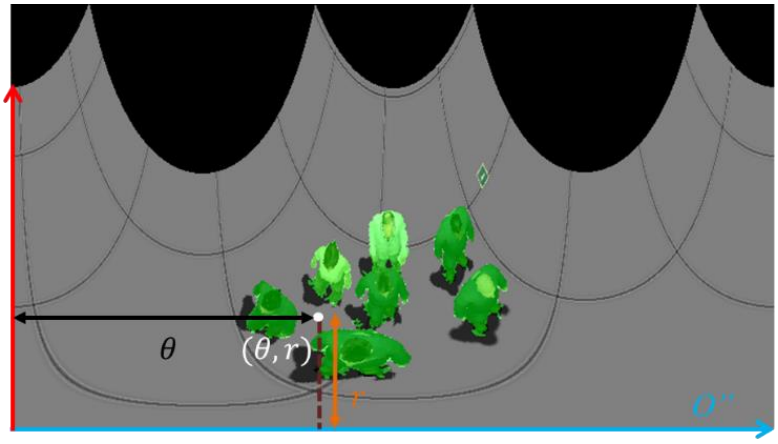
x and *y* in equations (2.7) and (2.8) are the coordinate values of an pixel in the original image coordinate system shown in Figure 2.23(a) where the image center O' is the origin of coordinates. r and θ are the coordinate values of the pixel after polar mapping and are rounded to integers to determine its location in the new image. The rounding can generate small 'holes' in the polar mapped image. To eliminate these 'holes', an inverse conversion is implemented. To get more details for the part far away from the image center, θ is sampled every 0.5 degree during the polar mapping, so there are 720 columns in the polar mapped image.

In Figure 2.23, the pixel (θ, r) in polar mapped image corresponds to the pixel (x, y) after polar mapping. θ and r denote the column number starting from the left and the row number starting from the bottom, respectively. The image center O' is converted as the bottom row, which is shown in blue in Figure 2.23(b).

If the image of a person is passed through by the positive x-axis shown as the red segment in Figure 2.23(a), it will be divided into two parts after polar mapping, with one part appearing in the far left and the other part in the far right of the polar mapped image. In this case, a circular shift on columns is implemented to combine the two parts.



(a) An image in the Cartesian coordinate system



(b) The image in the polar coordinate system

Figure 2.23: Polar mapping

Figure 2.24 shows a part of the image after performing polar mapping on the left image mentioned in Figure 2.23(b). The segment $\overline{P'Q'}$ along the vanishing line, as shown in Figure 2.22, becomes $\overline{P''Q''}$ that is in a column and has the same height l as $\overline{P'Q'}$, and A' is approximated by a rectangle A'' with the length being the height l and the width corresponding to the cylinder's diameter (average human width). So the image of a cylinder becomes a rectangle after polar mapping. l can be

evaluated by the number of foreground pixels right below P'', and the foreground area inside A'' can be computed using integral image [61-63] for the binary foreground blob.

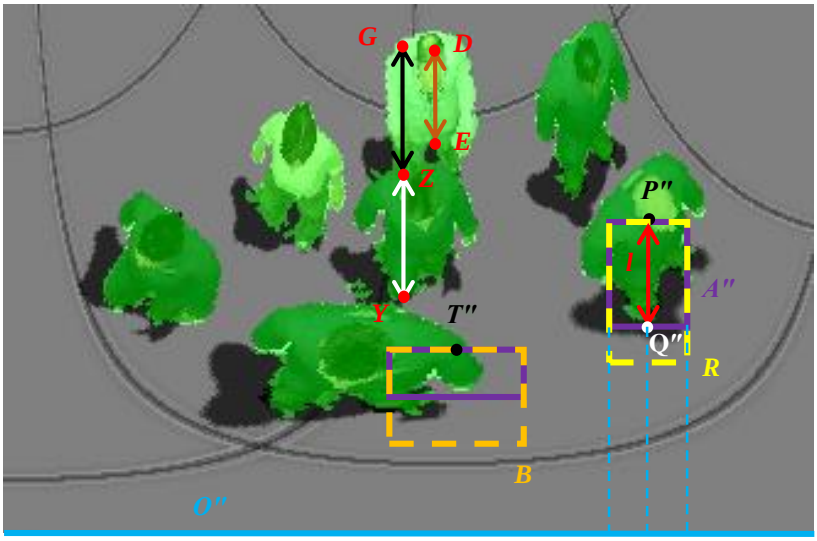
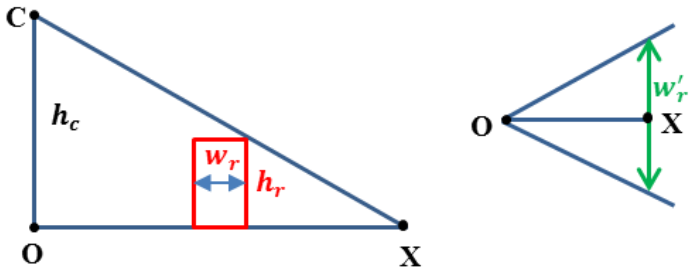


Figure 2.24: The reference projections and heights for foreground pixels

To evaluate the head area existence probability, the reference projected area for each foreground pixel is computed. It is defined as the projection of a reference person with an average human width w_r and height h_r assuming that the foreground pixel is the head point. The reference projection for a point is a rectangle since a person is modeled as a cylinder. In Figure 2.24, the dashed line rectangle R is the reference projection for the point P''.

Due to the perspective projection and polar mapping, the reference projected area varies with a pixel's location. For example, the dashed line rectangle B is the reference projection for point T'' and its size is different from R. So the height and

width of a reference projection are pre-computed for all pixels in the polar mapped image.



(a) A plane perpendicular to the ground (b) The ground plane

Figure 2.25: Pre-computing the height and width of the reference projection for a pixel in the polar mapped image

In Figure 2.25, C is a camera with height h_c; O is its projection on the ground plane, and X is a projected point on the ground corresponding to a pixel X'' in the polar mapped image. Figure 2.25(a) shows a plane perpendicular to the ground plane which is depicted in Figure 2.25(b). Assume that X'' is a head pixel, then the red rectangle in Figure 2.25(a) is the cross section of the reference person, and $w'_r = h_c \cdot w_r/(h_c - h_r)$ in Figure 2.25(b) is the projected width of w_r. The height and width of the reference projection for pixel X'', $\eta_h(X")$ and $\eta_w(X")$, are computed as

$$\eta_h(X") = h_r \frac{d(\overline{O"X"})}{h_c} + d(w_r) \tag{2.9}$$

$$\eta_w(X") = 2tan^{-1}(\frac{d(w'_r)}{2d(\overline{O"X"})})/\Delta\alpha \tag{2.10}$$

where $d(\cdot)$ denotes the distance in pixels and $\Delta\alpha$ is the sampling angle for polar mapping. $d(\overline{O''X''})$ is the row number of X'' from the bottom in the polar mapped image and O'' is image center after polar mapping, shown as the blue line in Figure 2.24. For example, $d(\overline{O''P''})$ is the row number of pixel P''. The number of pixels corresponding to a segment on the ground plane can be obtained by simple camera calibration.

The height of a foreground pixel X'', *H(X")*, is obtained by counting the number of consecutive pixels right below it in the same foreground blob. The height is calculated column by column from bottom to top. If *H(X")* is greater than $\eta_h(X'')$ or the pixel one row below X'' belongs to the background, the pixel counting restarts from 0 (this also infers that $H(X'') \leq \eta_h(X'')$). In Figure 2.24, $\overline{ED}$ denotes *H(D)* and the pixel counting starts from *E* because of the background pixels under *E*. *H(G)* is the length of $\overline{ZG}$ instead of $\overline{YG}$ since $H(Z) = \eta_h(Z)$. Thus occluded people can be roughly separated along a vanishing line (a column in the polar mapped image). The head area existence probability of a foreground pixel X'' in a polar mapped image is obtained by comparing with the pixel's reference projected area and is defined as

$$p_e(X'') = \frac{N(X'')}{\eta_h(X'')\eta_w(X'')} \tag{2.11}$$

where *N(X")* is the number of foreground pixels inside the rectangle with its top side centering at X'', width being $\eta_w(X'')$ and height being *H(X")*. Compared with *H(X")*, *N(X")* can determine a possible head pixel better, since a head pixel needs to be not only 'high' enough but also around the vertical central axis of a person. *N(X")* embeds the both features of a head point. In Figure 2.24, *H(G)* is greater than

H(D), but *G* does not necessarily have higher odds over *D* to be a head pixel as *D* is closer to the vertical central axis of the person. $p_e(P'') > p_e(T'')$, because the foreground inside A'' covers more area of the reference projection.

2.4.2 Potential Head Top Segment Detection

Figure 2.26: ROIs with color coded head area existence probabilities in the polar mapped image

We consider all pixels with the existence probability greater than a threshold t_p, not just those with the largest probability, as possible head area pixels. Since people are not perfect cylinders and a person's height and width can be somewhat different from the averages, the probabilities of the head area pixels of one person may be lower than those of another person. Some head area pixels will not be detected if just those with maximal probability are considered. The threshold t_p (set as 0.5 here) cannot be too high, since the computed existence probability of a true head area

pixel can be relatively low if a part of a person is occluded by another. The region formed by these detected possible head pixels is considered as the region of interest (ROI). In Figure 2.26, ROIs are shown as colored masks in a part of the polar mapped image, with the color indicating the head area existence probability. The pixels with high probabilities (those marked in red) are located not just in the head areas but also on the shoulders and around the necks. Thus the ROIs mainly lie on the top central part of a person.

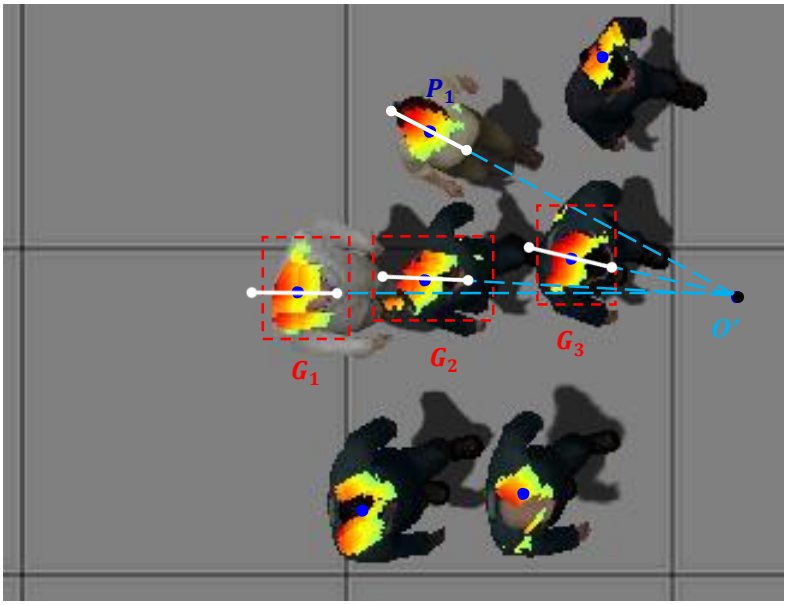


Figure 2.27: Establishing a potential head top segment

It's possible that more than an ROI exists in a foreground blob especially when it contains more than one person, thus ROI clustering is needed. To cluster the ROIs belonging to a person, the polar mapped image is converted back to the original one, as shown in Figure 2.27. ROIs with very small areas are considered not robust

and removed before clustering. The clustering starts by merging the largest ROI with any other ROI whose centroid is within an average shoulder width to the centroid of the largest ROI. The clustering continues with the largest ROI of the remaining ROIs until all the ROIs are clustered. The number of persons in a foreground blob can be estimated from the number of clusters. The three persons around the middle of Figure 2.27 are very close to each other and thus are detected in the same foreground blob within which the several ROIs are clustered into 3 groups, G_1, G_2 and G_3, corresponding to the three persons.

The centroids of the clustered ROIs, considered as the points of interest (POIs), are shown as blue dots in Figure 2.27. When a foreground blob contains the image center, it spreads over the entire rows after polar mapping, and the POI cannot be obtained using the aforementioned method. This happens when a person is very close to the FOV center. In this case, if the scene is not extremely crowded, he or she will not be occluded and the head area roughly locates at the center of the foreground blob, thus the ROI can be considered as the center part of the blob and the POI as the blob centroid. Since an ROI (in either case) appears not just in the head area, the POI is not necessarily located at the head top center. To get an accurate 3D head point, a potential head top segment is established by extending the POI to both sides by a short length (e.g., the diameter of the head top) along the vanishing line and only the part lying on the foreground is considered. In Figure 2.27, the POI P_1 is almost outside the head top but the potential head top segment (the white one) passes through the whole head top, and the head point can be found

on the segment. The potential head top segments for the three persons G_1, G_2 and G_3, the 3 white segments, are also illustrated.

Once the potential head top segments are detected, the head points are detected on the segments and their 3D positions are estimated using method in section 2.3.3.

2.4.3 Pedestrian Tracking

Pedestrian tracking is to associate a pedestrian's location across time/frames. In this section, tracking is performed by fusing 2 kinds of information. Pedestrians are first tracked based on the 3D head point detected using geometric cues. If the detected head point is not accurate, they are tracked based on the similar color distribution of a head top within two consecutive frames. Tracking based on the first kind of information is presented in section 2.3.4, so it is not introduced here. Tracking based on the second information and the combination of the 2 kinds of information is given as follows.

2.4.3.1 Tracking Based on the Color Cues of a Head Top

Many works [51, 64-67] use the intensity or color within an object region to form a template or histogram of the object in each frame. The correspondences of an object between two (usually consecutive) frames is established by evaluating the similarities of the two templates or color histograms. In the overhead view of a scene, the head top of a person is usually visible and the color distribution of a head top in two consecutive frames are very similar. A head top is considered as a flat circular region parallel to the ground and its size from an overhead view is irrelevant to the person's location. A color histogram is used as the representation

of a head top since it is invariant to scaling and rotation and can handle partial occlusions [66, 68].

The probability of the feature (color) *u* (*u=1, 2, …, m*) in a head top (the target) in the previous frame is computed as [66, 69]

$$q_u(\boldsymbol{y}) = C\sum_{i=1}^{n} k\left(\left\|\frac{\boldsymbol{x_i}-\boldsymbol{y}}{r}\right\|^2\right)\delta[b(\boldsymbol{x_i})-u] \tag{2.12}$$

$$C = 1/\sum_{i=1}^{n} k\left(\left\|\frac{\boldsymbol{x_i}-\boldsymbol{y}}{r}\right\|^2\right) \tag{2.13}$$

where $\{\boldsymbol{x_i}\}_{i=1,2,\dots n}$ are the locations of pixels inside the target region which has in total *n* pixels. $\boldsymbol{y}$ and r are the center and radius of the head top, respectively. $k(\cdot)$ is an isotropic kernel profile. C is the normalization function that is independent of $\boldsymbol{y}$ and can be pre-computed given $k(\cdot)$ and r. δ is the Kronecker delta function. $b(\boldsymbol{x_i})$ associates the pixel at location $\boldsymbol{x_i}$ to a specific histogram bin out of *m* bins. A head top centered at $\boldsymbol{y}$ is then modeled as $\boldsymbol{q}(\boldsymbol{y}) = \{q_u(\boldsymbol{y})\}_{u=1,2,\dots m}$.

Similarly, a candidate head top region (the target candidate) centered at location $\boldsymbol{v}$ and with the radius $h(h > r)$ in the current frame is represented as $\boldsymbol{p}(\boldsymbol{v}) = \{p_u(\boldsymbol{v})\}_{u=1,2,\dots m}$.

$$p_u(\boldsymbol{v}) = C_h\sum_{i=1}^{n_h} k\left(\left\|\frac{\boldsymbol{s_i}-\boldsymbol{v}}{h}\right\|^2\right)\delta[b(\boldsymbol{s_i})-u] \tag{2.14}$$

$$C_h = 1/\sum_{i=1}^{n_h} k\left(\left\|\frac{\boldsymbol{s_i}-\boldsymbol{v}}{h}\right\|^2\right) \tag{2.15}$$

where $\{\boldsymbol{s_i}\}_{i=1,2,\dots n_h}$ are the locations of pixels inside the target candidate that has n_h $(n_h > n)$ pixels and C_h is the normalization function. The radius h of the candidate region is determined by the average walking velocity and is usually smaller than 2m/s.

By using the aforementioned normalized histograms $\boldsymbol{q}(\boldsymbol{y})$ and $\boldsymbol{p}(\boldsymbol{v})$, the similarity between the head top and its candidate is evaluated based on the Bhattacharyya coefficient[70]:

$$\rho[\boldsymbol{q}(\boldsymbol{y}), \boldsymbol{p}(\boldsymbol{v})] = \sum_{u=1}^{m} \sqrt{q_u(\boldsymbol{y}) p_u(\boldsymbol{v})} \tag{2.16}$$

This coefficient has a geometric interpretation: the angle between the two vectors $\boldsymbol{q}(\boldsymbol{y})$ and $\boldsymbol{p}(\boldsymbol{v})$. The head point, i.e., the center of the head top, in the current frame can be detected by maximizing the Bhattacharyya coefficient in (2.16). The maximization is an iterative process and the search of the head point in the current frame is initialized with the head point position $\boldsymbol{y}$ in the previous frame. The search radius h is determined by the velocity of the target estimated from up to previous frame. By using the Taylor expansion around $p_u(\boldsymbol{y})$, $\rho[\boldsymbol{q}(\boldsymbol{y}), \boldsymbol{p}(\boldsymbol{v})]$is approximated as

$$\begin{aligned} \rho[\boldsymbol{q}(\boldsymbol{y}), \boldsymbol{p}(\boldsymbol{v})] \approx & \frac{1}{2} \sum_{u=1}^{m} \sqrt{q_u(\boldsymbol{y}) p_u(\boldsymbol{y})} \\ & + \frac{C_h}{2} \sum_{i=1}^{n_h} w_i k\left(\left\| \frac{\boldsymbol{s_i} - \boldsymbol{v}}{h} \right\|^2 \right) \end{aligned} \tag{2.17}$$

$$w_i = \sum_{u=1}^{m} \sqrt{\frac{q_u(\boldsymbol{y})}{p_u(\boldsymbol{y})}} \delta[b(\boldsymbol{s_i}) - u] \tag{2.18}$$

Algorithm 1: Tracking head points based on color histograms

Input: the head top model $\{q_u(\boldsymbol{y})\}_{u=1,2,\dots m}$ and its center $\boldsymbol{y}$ that is the head point in the previous frame

Output: the head point $\boldsymbol{y_1}$ and head top model $\{q_u(\boldsymbol{y_1})\}_{u=1,2,\dots m}$ in the current frame

1: Initialize the location of the head point in current frame with $\boldsymbol{y}$ and set iteration number $j = 0$

2: Compute the weights $\{w_i\}_{i=1,2,\dots n_h}$using (2.18)

3: Find the next location of candidate head point $\boldsymbol{y_1}$ using (2.20)

4: $d = \|\boldsymbol{y_1} - \boldsymbol{y}\|,\ \boldsymbol{y} = \boldsymbol{y_1}$

5: **if** $d < \varepsilon\ or\ j \geq N$, where ε is a threshold (default 1 pixel) and N is the maximum iteration number (default 10)

6: Compute the head top model $\{q_u(\boldsymbol{y_1})\}_{u=1,2,\dots m}$ in the current frame

8: Stop

9: **else**

10: *j=j+1* and go to step 2

Because the first term in (2.17) is independent of $\boldsymbol{v}$, we need to maximize the second term in order to get the maximum of (2.16). By employing the mean shift iteration [71], the estimated head point moves from current location $\boldsymbol{y}$ to the new location $\boldsymbol{y_1}$ calculated as

$$\boldsymbol{y_1} = \frac{\sum_{i=1}^{n_h} \boldsymbol{s_i} w_i g\left(\left\|\frac{\boldsymbol{s_i} - \boldsymbol{y}}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\boldsymbol{s_i} - \boldsymbol{y}}{h}\right\|^2\right)} \tag{2.19}$$

where $g(\cdot) = -k'(\cdot)$, assuming that the derivative of $k(\cdot)$ exists for all positive values. The kernel $k(\cdot)$ with Epanechnikov profile [71] is recommended to be adopted thus $g(\cdot)$ becomes constant and (2.19) is reduced to a simple weighted average [66]:

$$\boldsymbol{y_1} = \frac{\sum_{i=1}^{n_h} \boldsymbol{s_i} w_i}{\sum_{i=1}^{n_h} w_i} \tag{2.20}$$

The mean shift tracking algorithm is presented in Algorithm 1. The head point in the current frame is obtained using an iteration process and the iteration ends when the calculated head points in the previous and current iterations are close enough or the number of the iterations arrives at the preset threshold.

2.4.3.2 Tracking by Fusing the Geometric and Color Cues

To track a person in 3D, we can first detect all the 3D head points in the current frame using the projective geometric cues as mentioned previously in section 2.4.2, and then associate the detected or predicted head point with the corresponding person in the previous frame, as described in section 2.3.4. The head point detecting and associating are separated, and it's possible that both the detected and predicted head point is not accurate, shown as the white point in The mean shift tracking

algorithm is presented in Algorithm 1. The head point in the current frame is obtained using an iteration process and the iteration ends when the calculated head points in the previous and current iterations are close enough or the number of the iterations arrives at the preset threshold.(b). In this case, the corresponding head point in the current frame can be determined using the color information of the head top that is shown as the region inside the red circle with the head point as the center in The mean shift tracking algorithm is presented in Algorithm 1. The head point in the current frame is obtained using an iteration process and the iteration ends when the calculated head points in the previous and current iterations are close enough or the number of the iterations arrives at the preset threshold.(a). Thus the detecting and associating (together, called tracking) are performed at the same time.

(a) Part of the previous frame (b) The same region as (a) in the current frame

Figure 2.28: A head point in the current frame (b) is detected using the projective geometry and associated with the head point in the previous frame (a) based on the constant velocity within 2 consecutive frames. The frames in (a) and (b) are from the left overhead camera

Once the 2D head point is detected, its 3D position can be calculated via triangulation after finding the matching point in the right image. However, to track the head point based on the color histogram of the head top using mean shift as in section 2.4.3.1, an initial head point such as the white point in The mean shift tracking algorithm is presented in Algorithm 1. The head point in the current frame is obtained using an iteration process and the iteration ends when the calculated head points in the previous and current iterations are close enough or the number of the iterations arrives at the preset threshold.(a) is needed. The head point detected based on the projective geometry and tracked based on constant velocity within two successive frames is a good choice for the initialization. Thus the geometric cues (together with the common motion assumption) and color cues are two complementary tools for pedestrian tracking and are integrated to improve the robustness and accuracy of the tracking results. The new tracking algorithm is given as in Algorithm 2.

Line 3 in Algorithm 2 determines whether the detected head point using the geometric cues is accurate or not by evaluating the similarity of the color histograms within the two red circles shown in The mean shift tracking algorithm is presented in Algorithm 1. The head point in the current frame is obtained using an iteration process and the iteration ends when the calculated head points in the previous and current iterations are close enough or the number of the iterations arrives at the preset threshold.. If the head point detected based on the color cues is an outlier, i.e., the detected 2D head point lies in the background or the height of the 3D head point changes drastically, as shown in Line 7, the head point detected

based on the geometric cues is still adopted. This procedure is used as a cross validation and thus can improve the tracking robustness.

Algorithm 2: Tracking head points based on the common motion assumption and color cues

---

Input: the 2D, 3D head point $\boldsymbol{X_{2d}}$, $\boldsymbol{X_{3d}}$ of a person *f* and the color distribution of the head top $\boldsymbol{q}(\boldsymbol{X_{2d}})$ in the previous frame

Output: the 2D, 3D head point $\boldsymbol{Y_{2d}}$, $\boldsymbol{Y_{3d}}$ of the person *f* and the color distribution of the head top $\boldsymbol{q}(\boldsymbol{Y_{2d}})$ in the current frame

1: Detect all the head points based on geometric cues and find the 2D and 3D head point, $\boldsymbol{Y_{2d}}$ and $\boldsymbol{Y_{3d}}$, for the person *f* based on constant moving velocity in current frame

2: Compute the color histogram of the head top $\boldsymbol{q}(\boldsymbol{Y_{2d}})$

3: **if** $\rho[\boldsymbol{q}(\boldsymbol{Y_{2d}}), \boldsymbol{q}(\boldsymbol{X_{2d}})] > \varepsilon_1$(a threshold, set as 0.8)

4: Stop

5: **else**

6: Find the 2D head point $\boldsymbol{T_{2d}}$ for person *f* based on $\boldsymbol{q}(\boldsymbol{X_{2d}})$ using mean shift and calculate its 3D location $\boldsymbol{T_{3d}}$

7: **if** $\boldsymbol{T_{2d}}$ is in the foreground and *z*-value of $\boldsymbol{T_{3d}}$ is within a reasonable range

8: $\boldsymbol{Y_{2d} = T_{2d}, Y_{3d} = T_{3d}}$

9: **else**

10: Stop

2.4.4 Experiments

2.4.4.1 Experiment Setup

As in the uncrowded scenes in section 2.3, we test our approach using a publicly available visual surveillance simulation test bed, ObjectVideo Virtual Video (OVVV) [57].

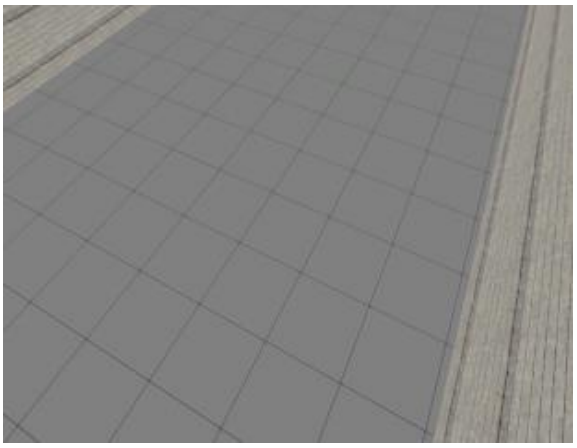


Figure 2.29: A virtual train station concourse

A virtual scene of a train station concourse with a flat ground as shown in Figure 2.29 is created, where two group people walk towards mainly two different directions. There are 14 people walking in an area of about 4*4.5 m, which is a crowded scene. The foreground blobs of people merge from time to time even in the overhead view of a scene. The ceiling is 8.84 m high from the ground. Two

identical horizontally aligned cameras are installed on the ceiling with the image planes parallel to the ground plane. The frame rate is 15 frames per second and the frame size is 640*480 pixels.

2.4.4.2 Experiment Results

Figure 2.30 shows two frames captured by the left camera. In these frames, people are close to each other thus occlusions occur. The green dots are the 2D head points detected based on the projective geometry and they are updated using mean shift if the color distributions within the head tops around the detected head points are not close enough to those in the previous frame. The corrected head points are shown as white dots in Figure 2.30. We can see that the green dots which are relatively further from the head top centers are updated by the white dots which are closer to the head top centers. Thus the detection of the head points becomes more accurate.

To better show the tracking results, the estimated 3D tracks are projected onto the X-Y plane (ground) and Z plane (height) separately. Figure 2.31(a) depicts the ground plane tracking results when the 3D head points are detected only using geometric cues and tracked based on the common motion assumption. Figure 2.31(b) shows ground plane tracking results when both the geometric and color cues are applied. In both figures 2.31(a) and (b), the solid and dashed lines of the same color represent the ground truth and estimated trajectory of the same person, respectively. The brown dots are the FOV centers. The number at one end of each trajectory denotes the object ID. The estimated trajectories are oscillating around

but very close to the ground truth, and the estimated tracks in Figure 2.31(b) are even closer to the ground truth.

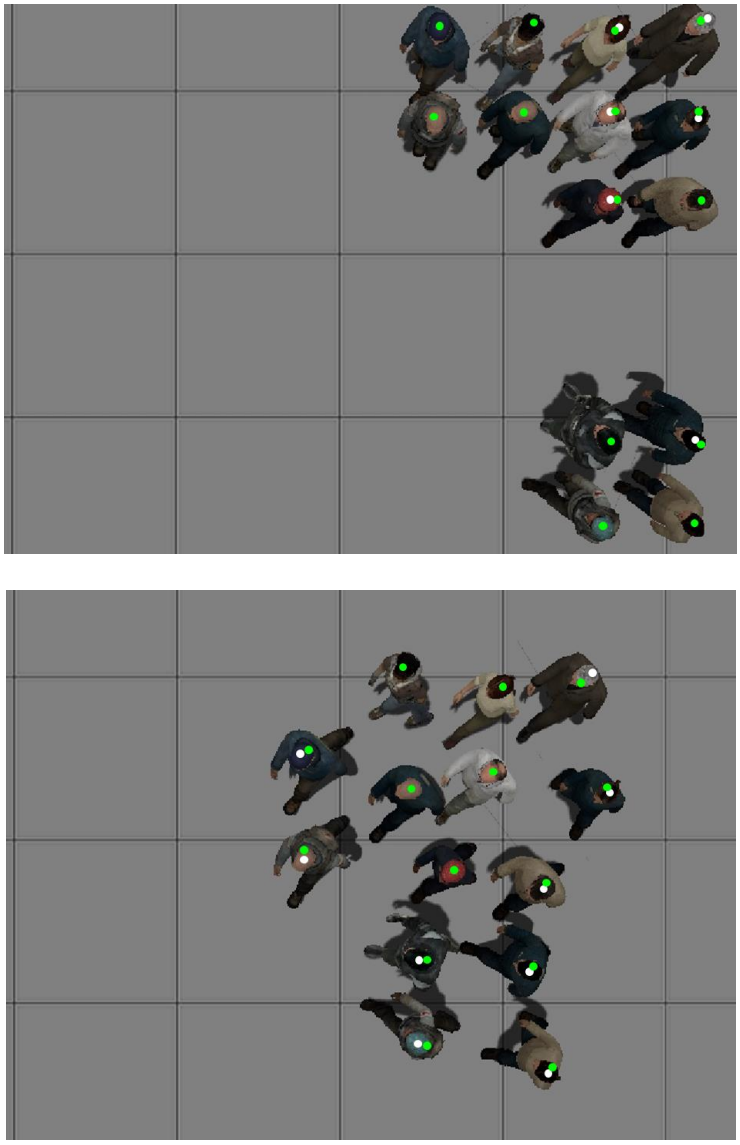
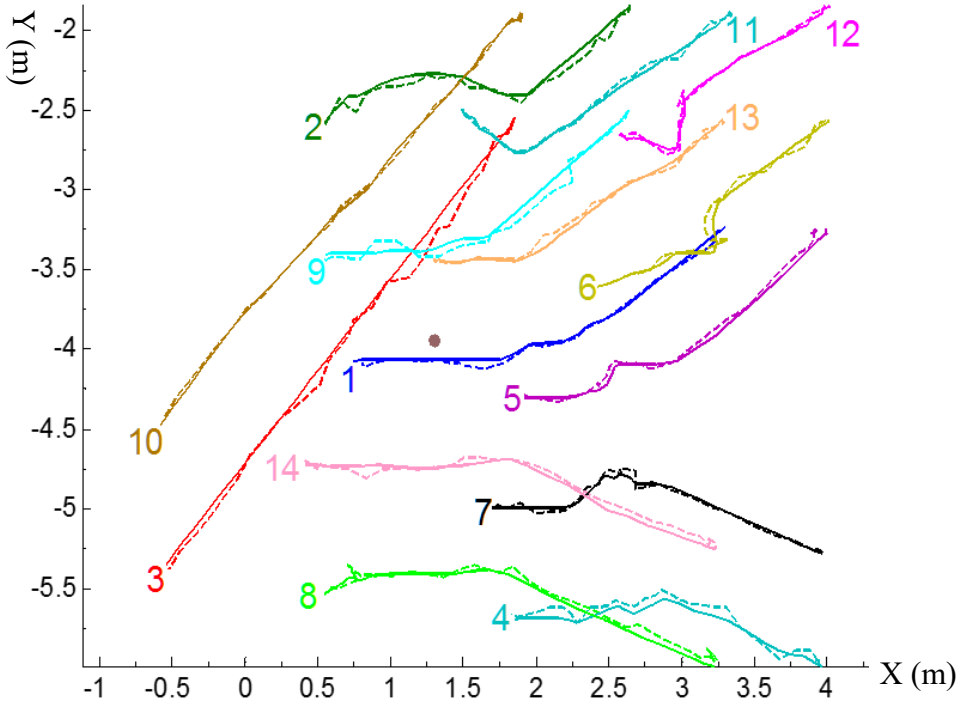
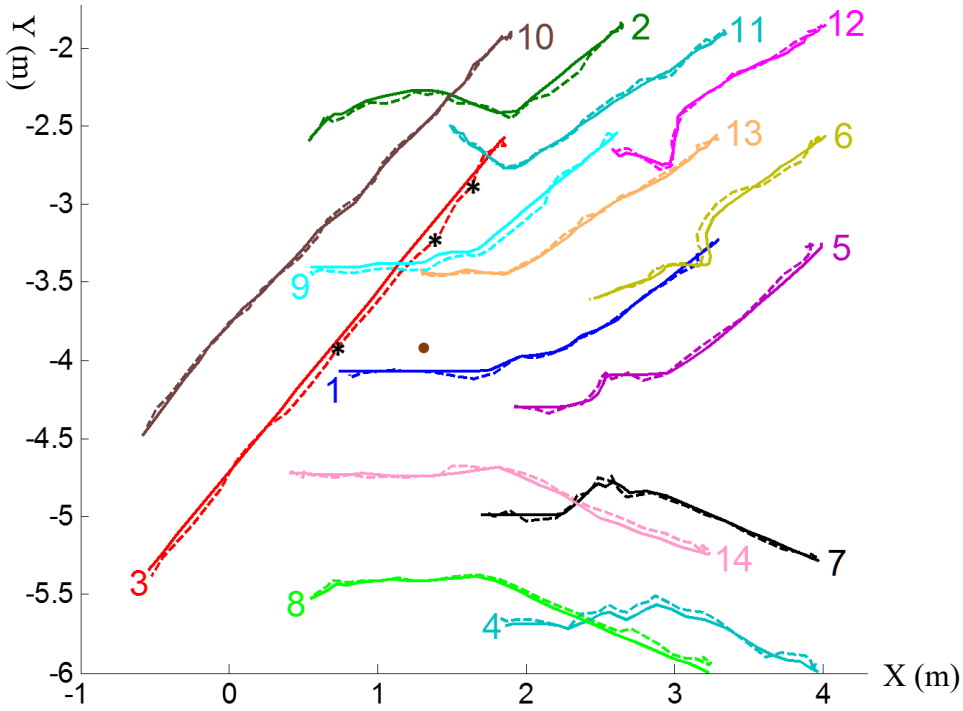


Figure 2.30: Two frames captured by the left camera with the head points detected by combining the geometric and color cues



(a) Tracking results only based on the geometric cues



(b) Tracking results based on both the geometric and the color cues

Figure 2.31: The ground plane tracking results

Table 2.4: The smallest and largest four average errors of the estimated tracks when only the geometric cues are used

Object ID	Ground Plane Errors (cm)	Height Errors (cm)	3D Errors (cm)
11	2.40 ± 1.28	2.07 ± 0.62	3.31 ± 1.06
13	2.48 ± 1.59	2.21 ± 1.31	3.42 ± 1.90
12	2.83 ± 1.83	2.50 ± 1.96	3.89 ± 2.51
1	3.15 ± 2.32	4.07 ± 2.62	5.41 ± 3.07
8	4.90 ± 2.50	3.87 ± 2.54	6.36 ± 3.35
3	5.06 ± 2.47	3.56 ± 3.94	6.59 ± 4.03
4	5.30 ± 2.05	3.87 ± 2.74	6.83 ± 2.82
6	5.67 ± 1.72	3.95 ± 3.90	7.18 ± 3.76

Table 2.5: The smallest and largest four average errors of the estimated tracks when both the geometric and color cues are used

Object ID	Ground Plane Errors (cm)	Height Errors (cm)	3D Errors (cm)
13	2.23 ± 1.16	1.91 ± 0.67	3.04 ± 1.09
12	2.53 ± 1.29	1.95 ± 0.70	3.30 ± 1.22
11	2.52 ± 1.35	2.03 ± 0.70	3.89 ± 1.09
2	2.72 ± 1.81	2.17 ± 0.63	4.23 ± 1.41
4	4.69 ± 1.97	2.83 ± 2.28	5.74 ± 2.45
3	5.21 ± 1.94	2.24 ± 0.82	5.83 ± 1.57
9	4.90 ± 1.62	2.90 ± 2.18	5.91 ± 2.18
6	5.54 ± 1.13	3.32 ± 1.64	6.55 ± 1.63

Table 2.6: The overall tracking errors when only the geometric cues are used

Ground Plane Errors (cm)	Height Errors (cm)	3D Errors (cm)
4.02 ± 2.38	3.08 ± 2.80	5.34 ± 3.25

Table 2.7: The overall tracking errors when both the geometric and color cues are used

Ground Plane Errors (cm)	Height Errors (cm)	3D Errors (cm)
3.76 ± 2.16	2.65 ± 1.93	4.86 ± 2.44

The tracking errors are reported as the average distances between the true and estimated positions across all frames for each object. The smallest and largest four average 3D tracking errors are tabulated in Table 2.4 and Table 2.5 together with the ground plane and height tracking errors. Table 2.4 and Table 2.5 show the tracking errors when one and two tracking cues are used, respectively. Table 2.6 and Table 2.7 show the corresponding overall tracking errors. From the tables, it's obvious that tracking based on fusing two tracking cues gives better tracking results. To the best of our knowledge, the smallest error of the ground plane trajectories in others' works is around 5 cm as reported in paper [37] where more than 2 side view cameras are used and the scene is sparse.

From Table 2.4 to Table 2.7, we can see that the ground plane, height and 3D tracking errors are dependent. When estimating the 3D head position, the height of person is first calculated, and based on which the ground plane coordinates are obtained. The 3D head position errors mainly result from the fact that sometimes the estimated potential head top segment can be slightly off the head top center. Pedestrians walking in the scene are not perfect cylinders, hence the foreground blobs are not completely symmetric about the vanishing line that passes through the head top center. Although the detected head point might miss the head top center, it still lies inside the head top. This results in an error on the X-Y plane that is

usually smaller than the head top radius (about 8 cm on average for adults). Even if the detected head point misses the head top center, the estimated height is very close to the true value since the head top is relatively flat. This is also validated in Table 2.4 to Table 2.7 where the height errors are usually smaller than the corresponding ground plane errors.

To demonstrate that our accurately estimated 3D head positions can be helpful in face tracking and recognition, a PTZ camera with a height of 4 m, ground plane coordinate (-1, -9) m (the same coordinate system shown in Figure 2.31(b)) and a resolution of 320*240 pixels is installed on the wall. The FOV of the PTZ camera is set small (1.27*0.95 m) to capture high resolution close-up facial images. If the 3D head position is not accurate enough, no or only partial facial image can be captured. The pan and tilt angles of the PTZ camera are determined by the ground plane position and the height of the target, respectively. The PTZ camera is guided by the 3D trajectories obtained by fusing two geometric and color cues. In Figure 2.32, the three close-up facial images from left to right are captured when person 3 arrives at the locations marked by the asterisks in Figure 2.31(b). The arrow denotes the walking direction. Our method provides accurate 3D head locations and thus is very effective in capturing close-up facial image, with almost all the captured faces around the image center. Even for the second capture location where both X-Y and Z plane errors are relatively big, the whole face is still captured (the middle image in Figure 2.32). To always capture the frontal face images for better recognition, more than one PTZ cameras are required thus a camera scheduling is also needed. This is presented in next chapter.

Figure 2.32: Close-up facial images of person 3 captured by the PTZ camera

2.5 Conclusions

In this chapter, we detect and track pedestrians in an indoor environment from an overhead view based on 3D head points. We start with uncrowded scenes where people are not occluded and their images are not connected from an overhead view. In uncrowded scenes, a potential head top segment is detected for each person directly from projective geometry. To obtain a head top segment in crowded scenes, possible head areas are determined inside each foreground blob by evaluating the head area existence probability based on projective geometry. The highest points on the segment are detected to estimate the 3D head position efficiently. To track a person, the detected 3D head point is associated with the corresponding head point in the previous frame based on the common motion assumption. In crowded scenes, when the head point detected using the geometric cues is not accurate enough, it is updated using color cues of the head top region across consecutive frames. The approach is tested using a publicly available visual surveillance simulation test bed. The experiments show that the 3D tracking errors are very small under both crowded and uncrowded scenes from overhead views. The tracking errors for

ground positions and heights are around 4 cm and 3 cm, respectively. We also demonstrate that our method can help PTZ cameras to capture close-up face images. Currently our tracking approach can't handle outdoor environment well since the background is more cluttered. By using a more clutter-tolerant foreground segmentation technique (e.g., extracting foreground objects through modeling the color of each pixel using a mixture of Gaussians), our method can also track people in outdoor environment.

3. SCHEDULING PTZ CAMERAS FOR FACE IMAGE CAPTURE BASED ON 3D PEDESTRIAN TRACKS

The 3D position of a person acquired in last chapter is very helpful to capture close-up face images for recognition, and frontal face images usually give good recognition results. To achieve this, a smart camera network containing 2 overhead cameras and a set of PTZ cameras is constructed. With the detected 3D head position, we pick a most appropriate PTZ camera to capture the best frontal face images of a person of interest across time. A PTZ camera is selected based on the frontal face capture quality which is measured by the head visibility from the PTZ camera (a factor decided by both the target and other people in the scene and rarely considered by other researchers), view angle of the frontal face, camera-face distance and mechanical limits of the PTZ camera. When a new PTZ camera is chosen, the handoff success probability of the two PTZ cameras (the old and new one), which reflects the response time of the new camera moving from the initial to desired state, is also taken into account to capture close-up face images seamlessly. Experiments are implemented in a publicly available visual surveillance simulation test bed and show that our approach can capture the high-quality frontal face images effectively.

3.1 Introduction

To observe large scenes, cameras always have a wide field of view and are installed from a distance. This makes tasks that are sensitive to image quality and resolution, such as vehicle license plate recognition, face identification and gesture

recognition, very difficult. To get the details of objects clearly, PTZ cameras that can zoom in and adjust their orientations to follow/focus on the objects across time are needed.

Using PTZ cameras together with fixed cameras to capture high quality videos of interested targets have been concerned by many researchers. Collins et al. [1] present a master-slave sensor corporation form with a wide-angle view camera as the master and a highly zoomed in pan-tilt camera as the slave. Object geoloccations are estimated from the master camera's view point by intersecting back-projected viewing rays with a terrain model. Then a pan-tilt command transformed from the location is sent from the master camera to obtain the close-up image of the object. Zhou et al. [72] also use one master camera and one slave camera to acquire biometric images of humans for recognition. They sample some pixels related to actual points in a surveillance scene and record the pan-tilt angles by which the slave camera can center on the points. For every other pixel on the master image, the slave camera angles are calculated by interpolation of the previously recorded angles. Instead of being controlled completely by the master camera, the slave camera only uses the pan-tilt command initially. After the target detection is achieved in the slave view, the slave camera starts to track the target. The system is designed to detect and track only one object. Stillman et al. [73] use two static and two PTZ cameras to track and recognize at most two people that wear clothes of different colors. The world coordinate obtained from the triangulation of the target locations from two static cameras provides a coarse estimate of the pan-tilt angle and zoom factor. Like paper [72], PTZ cameras start tracking based on

their video stream once locking on the target. But they also use the information from static cameras to guide PTZ cameras when PTZ cameras lose the targets.

A PTZ camera is assigned to a single target for a certain period of time to record high quality images of all targets. The assigning problem is not concerned if there is only one PTZ active camera and one target being followed [72, 74]. But if there is more than one PTZ camera and/or one target, the scheduling problem becomes increasingly non-trivial. Costello et al. [75] evaluate various strategies for scheduling a single PTZ camera to acquire biometric imagery of the people present in the scene. Hampapur et al. [76] present several ways to assign cameras to the subject being monitored, such as location-specific assignment which assigns active cameras to the objects within certain area, round robin sampling which assigns cameras to different objects periodically to achieve uniform coverage, etc. But how to implement those methods is not mentioned, since their experiment is based on only one target. Instead of assigning equal importance to each object, Qureshi and Terzopoulos [59] propose a weighed round robin scheduling algorithm to capture high quality videos for pedestrians based on their arrival time and the suitability of an active camera with respect to focusing on a pedestrian. Bimbo and Pernici [77] rank the importance of objects by evaluating the estimated deadline by when they will leave the scene and take into account of the cost of camera movements to acquire close-up images of as many targets as possible. Krahnstoever et al. [78] build a probabilistic objective function parameterized on the capture distance, view angle, target-zone boundary distance and mechanical limits of PTZ cameras for each target. By maximizing the objective function, a balance between the number

of captures per target and their quality is attained. To avoid camera scheduling which may appear optimal at present but will later lead to observation and tracking failures, Qureshi and Terzopoulos [79] apply a camera assignment considering both the short-term and long-term effects. With a scene under surveillance becoming crowded, occlusions caused by other people in the scene are prone to happen in the close-up images captured by PTZ cameras, which is rarely considered in most existing camera scheduling methods.

In this chapter, a set of PTZ cameras are used to capture close-up face images of a target with the guidance of the 3D head positions obtained by fixed overhead cameras, as shown in Figure 3.1. A PTZ camera that captures the best frontal close-up face images of the target, shown as the person in the circle in Figure 3.1, is called the best PTZ camera and is selected to capture face images across time. The best PTZ camera is determined not only by the target and candidate PTZ cameras but also other people in the scene who may block the view of target in close-up images.

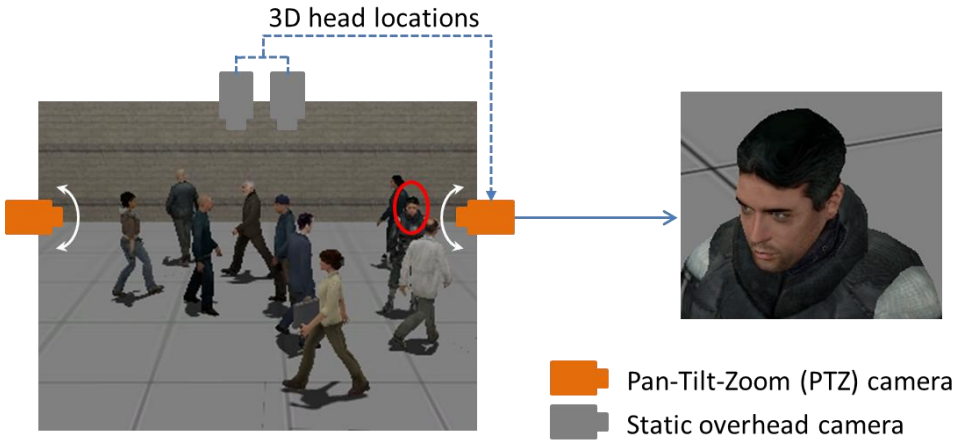


Figure 3.1: A camera network for close-up face image capture

3.2 Best PTZ Camera Selection based on Capture Quality

The best PTZ camera varies with the location and velocity of the target. It is selected based on the frontal face capture quality that is evaluated by the head visibility, view angle of the frontal face, camera-face distance and camera mechanical limits.

3.2.1 Capture Quality Measures

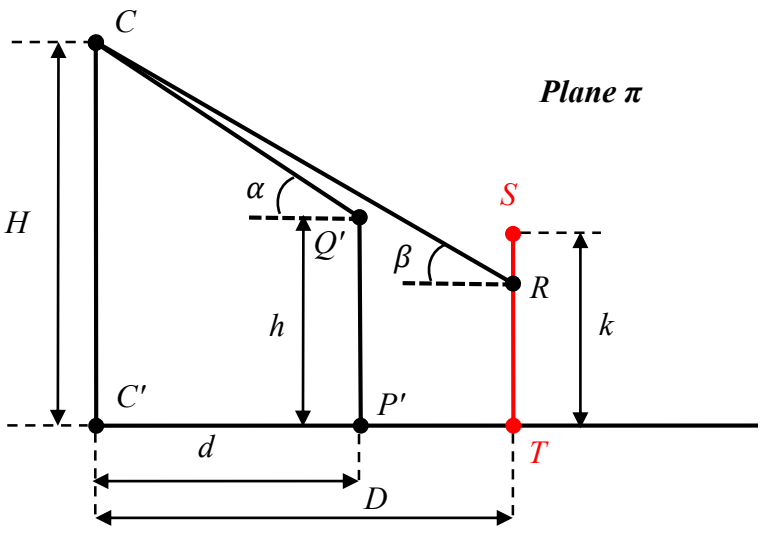
In this section, quantitative models are built to measure the frontal face capture quality of each PTZ camera.

3.2.1.1 Head Visibility

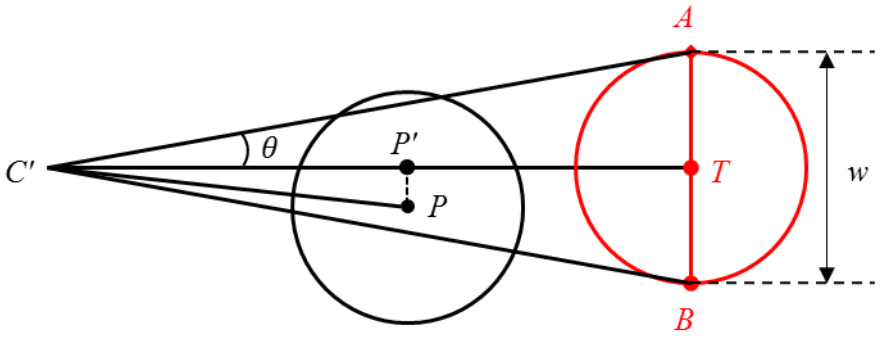
To capture a face image, the head of a person of interest needs to be visible from the PTZ camera. When a scene becomes crowded, the head of the target can be blocked by other people in the scene from the point of view of a PTZ camera. If this happens, the PTZ camera will not be picked. Hence we need to consider the impact of other people on the PTZ camera selection. A person is modeled as a cylinder as shown in Figure 3.2, where C is a PTZ camera installed at a height of H from the ground and C' is the projection on the ground. ST is the central vertical axis of the person of interest, i.e., our target. Figure 3.2(a) shows the plane π that is formed by CC' and ST and thus is perpendicular to the ground. $Q'P'$ on the plane π is the projection of QP which denotes a person that may obstruct the target. SR is the head part of the target and the length of RT is γk, where k is height of the target and $\gamma = \frac{65}{75}$ (since an average person is generally 7-and-a-half heads tall). Figure 3.2(b) shows the ground plane where P and T is the ground point of the

corresponding persons in Figure 3.2(a) and *w* is the average human width. The face of the target is blocked if that *CR* is intercepted by *Q'P'*, *P* is closer to *C'* and *P* is inside the triangle *AC'B* formed by *C'* and the target. The face visibility from the PTZ camera, i.e., the target's obstruction level to the camera, is quantified as

$$Q_{hv} = \begin{cases} 0 & if \angle TC'P < \theta \;\&\; C'P < C'T \;\&\; \alpha \leq \beta \\ 1 & otherwise \end{cases} \tag{3.1}$$



(a) A plane that is perpendicular the ground plane



(b) Ground plane

Figure 3.2: Evaluating the head visibility of a person of interest from the view of a PTZ camera

3.2.1.2 View Angle of the Frontal Face

Out of all kinds of views of a face, the frontal view can generate best face recognition results. Thus the view angle of the frontal face from a PTZ camera is crucial for high-quality face image capture. We assume people always face to the walking direction, i.e., the normal of the face and the walking direction are the same, which is a reasonable assumption. In Figure 3.3, the person walks following the arrow and δ (in degrees) is the view angle of the PTZ camera. The capture quality based on the view angle of the frontal face is measured by

$$Q_{va} = e^{-\frac{\delta^2}{2\sigma_{va}^2}}, \delta = [0,180) \tag{3.2}$$

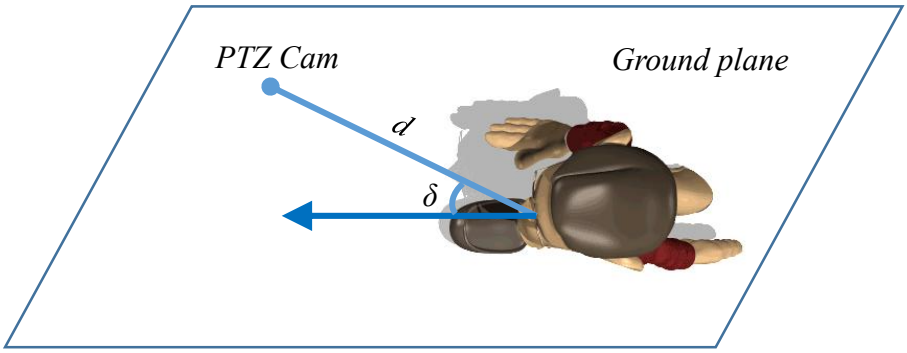


Figure 3.3: An overhead view of a person walking on the ground

The smaller the value of σ_{va}, the faster Q_{va} decreases when the target doesn't face to the PTZ camera. If $\sigma_{va} = 45$ degree and the PTZ camera captures the side view of the face, i.e. $\delta = 90$ degree, we will have $Q_{va} = 0.135$ which means low capture quality of face image.

3.2.1.3 Camera-target Distance

The quality of the captured face images generally degrades with the increase of the distance between the PTZ camera and the target. So the third quality measure is based on the camera-target distance given as

$$Q_d = \rho + (1-\rho)e^{-\frac{d^2}{2\sigma_d^2}} \tag{3.3}$$

where d is distance between the face and the PTZ camera, as shown in Figure 3.3. ρ represents a baseline capture quality of face image when the target is far away from the camera. The second term of the right side of equation (3.3) models the trend how the capture quality drops as the target moves away from the camera. σ_d is decided by the size of the scene monitored by the fixed overhead cameras and the distance from the center of the scene to the PTZ cameras.

3.2.1.4 Mechanical Limits

A PTZ camera has a mechanical limitation on the range of the pan, tilt and zoom parameters. It's impossible to set the parameter state of PTZ cameras out of the physical range to capture the face images. Hence, a term that defines the mechanical limits of a PTZ camera is introduced:

$$Q_m = \begin{cases} 1 & if\,(\emptyset, \psi, r) \in [\emptyset_{min}, \emptyset_{max}], [\psi_{min}, \psi_{max}], [r_{min}, r_{max}] \\ 0 & otherwise \end{cases} \tag{3.4}$$

where $[\emptyset_{min}, \emptyset_{max}]$, $[\psi_{min}, \psi_{max}]$ and $[r_{min}, r_{max}]$ are the mechanical limits of the pan, tilt and zoom parameters respectively.

3.2.2 PTZ Camera Selection

The success probability of capture P_{cap}, i.e., the probability that a PTZ camera captures a close-up frontal face image is evaluated using the quantitative measures in section 3.2.1, given as

$$P_{cap} = Q_{hv}Q_{va}Q_dQ_m \tag{3.5}$$

The capture probability reflects not only the status of the target and PTZ cameras but also other people in the scene. The PTZ camera generating the maximum probability is the best PTZ camera and is picked to follow the target. Assume that C_{prv} is the PTZ camera used in the previous frame and camera C_{best}, which is different from C_{prv}, is the best PTZ camera based on P_{cap} in the current frame. The pan-tilt-zoom parameter state S_{est} of C_{best} is estimated accordingly. If the current parameter state S_{cur} of C_{best} is very different from S_{est} and camera C_{best} is still used to capture the target, we may lose the target since C_{best} cannot arrive at the estimated parameter state in time. In this case, we need to stick with camera C_{prv} to capture the target with its corresponding estimated parameters, as shown in Figure 3.4. In the meantime, camera C_{best} keeps moving towards the estimated orientation and zooming state. Since pedestrians walk smoothly, if the same PTZ camera is used between two frames, usually it can change from the current parameter state to the estimated one in time and capture the target successfully. Hence, when a new PTZ camera C_{best} is selected based on P_{cap}, we need to consider the cost of changing from its current parameter state to the estimated state. It is measured by the handoff success probability, given as

$$P_h = e^{-Max\left(\frac{(\emptyset_{cur}-\emptyset_{est})^2}{2\sigma_{\emptyset}^2}, \frac{(\psi_{cur}-\psi_{est})^2}{2\sigma_{\psi}^2}, \frac{(r_{cur}-r_{est})^2}{2\sigma_r^2}\right)} \quad (3.6)$$

where $(\emptyset_{cur}, \psi_{cur}, r_{cur})$ and $(\emptyset_{est}, \psi_{est}, r_{est})$ are the current and estimated parameter sets respectively. σ_{ϕ}, σ_{ψ} and σ_r are determined by the panning, tilting and zooming speed of the PTZ camera and the frame rate of the fixed overhead cameras. A new PTZ camera C_{best} is used to capture the target only when P_h is greater than a threshold *t*, as shown in Figure 3.4. For example, the panning and tilting speed of a PTZ camera are 180 degrees/s; the zooming speed is 60 units/s; and the frame rate is 20 frames/s. In this case, the panning and tilting angle are at most 9 degrees and the difference of the zooming factor is up to 3 units within two frames. We set $\sigma_{\phi} = 9$, $\sigma_{\psi} = 9$ and $\sigma_r = 3$. When $|\emptyset_{cur} - \emptyset_{est}| < 9$, $|\psi_{cur} - \psi_{est}| < 9$ and $|r_{cur} - r_{est}| < 3$, we have $P_h > 0.6$. Thus 0.6 can be set as the threshold *t*.

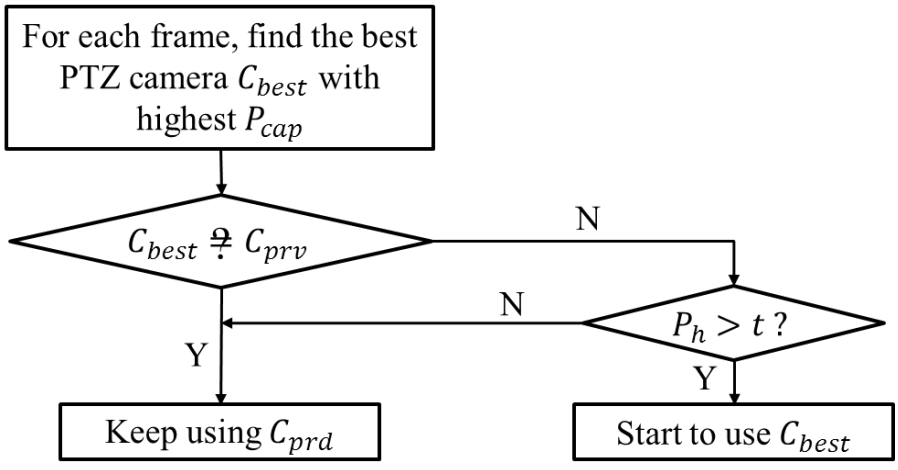


Figure 3.4: Selecting a most appropriate PTZ camera to capture frontal face images

3.3 Experiments

We test our PTZ camera scheduling approach using a publicly available visual surveillance simulation test bed, ObjectVideo Virtual Video (OVVV) [57]. A virtual train station concourse created in Chapter 2 is adopted and the two overhead cameras are installed in the same way as that in Chapter 2.4. 4 PTZ cameras with the resolution of 320*240 pixels and FOV of 1.27*0.95 m are installed around the scene, as shown in Figure 3.5 where the 4 blue dots denote the 4 PTZ cameras with the same height of 4.5 m and the image is the FOV of the left overhead camera. A person walks following the black arrows in the scene and the 3D head positions are obtained by fusing both the geometric and color cues of the person as present in Chapter 2.4.

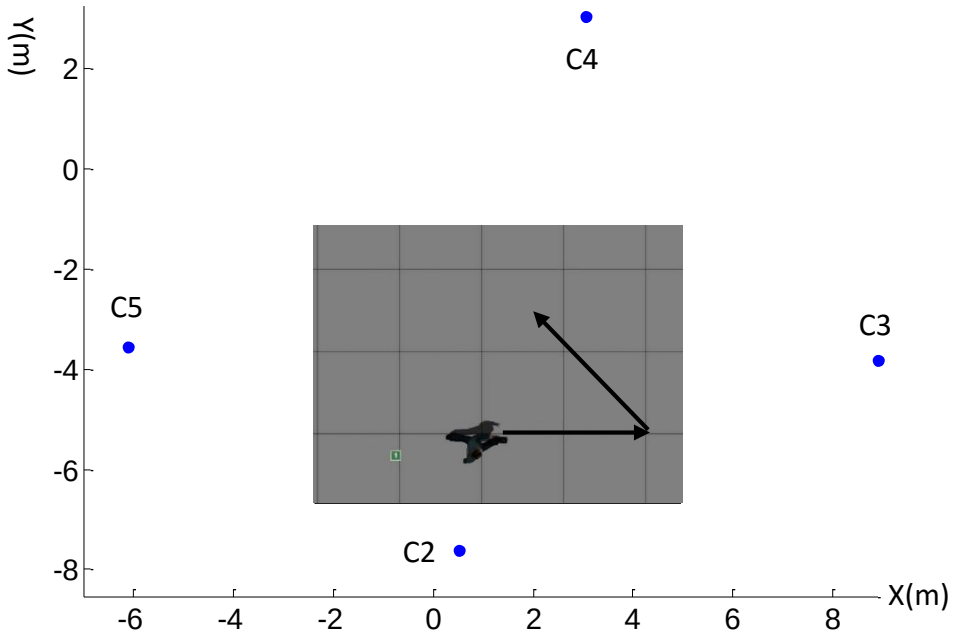
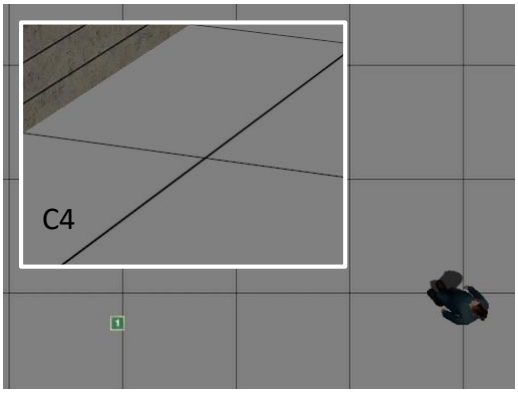


Figure 3.5: PTZ camera distribution

(a) Frame 50

(b) Frame 70



(c) Frame 108

(d) Frame 120

(e) Frame 140

Figure 3.6: Close-up face image capturing results when the handoff success probability is not considered

Figure 3.6 shows the close-up face image capturing results of the person when the handoff success probability of 2 PTZ cameras is not considered. The face images and the PTZ cameras used for capture are highlighted inside the white rectangles. PTZ Camera *C3* is first used to follow the target. The person changes her walking direction around frame 108 starting from which camera *C4* is selected to capture the face images based on the success probability of capture. From Figure 3.6(c), the face image is not captured by *C4* since there is much difference between the current and estimated parameter states at frame 108. *C4* is still on the transition stage to the estimated orientation and zooming state.

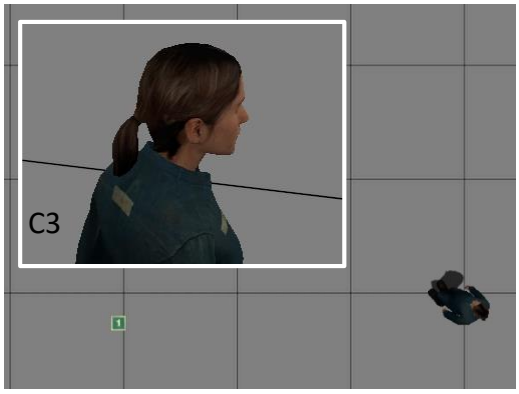


Figure 3.7: The face image captured at frame 108 when the handoff success probability is considered

If we consider the difference between the current and estimated parameters by evaluating the handoff success probability, the previously used camera *C3* is selected at frame 108 and the face image is captured, as shown in Figure 3.7. Although it is more like a side view face image, it's better than completely losing

the target. Meanwhile, *C4* keeps adjusting its parameters towards the estimated state. So, in the next few frame, the handoff success probability will be higher and higher and *C4* will be used to capture the face images. In our experiment, *C4* is assigned to follow the target starting from frame 110 when the handoff success probability is taken into account.

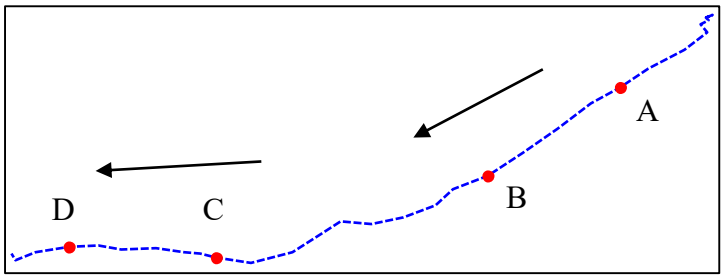


Figure 3.8: 4 face image capturing locations on the estimated track of person 1

We also test our method in a crowded scene mentioned in chapter 2 and the same PTZ camera distribution is employed as shown in Figure 3.5. Both the success probability of capture and the handoff success probability are considered as the scheduling scheme illustrated in Figure 3.4. Figure 3.8 shows the detected track of person 1 in the scene. Figure 3.9 (a) and (b) show the captured clos-up face images when person 1 arrives at the points A, B, C and D shown as the red dots in Figure 3.8. The overhead views of the scene when the face images are captured are illustrated in Figure 3.9(c) and (d) where person 1 is marked in the red circles. PTZ camera *C2* is the best PTZ camera to capture the face images when person 1 is at locations A and B. Person 1 changes her direction as shown in Figure 3.8 to avoid

(a) Face images captured when person 1 arrives at A (left) and B (right) in Fig 3.8

(b) Face images captured when person 1 arrives at C (left) and D (right) in Fig 3.8

(c) Overhead views when person 1 arrives at A (left) and B (right) in Fig 3.8

(d) Overhead views when person 1 arrives at C (left) and D (right) in Fig 3.8

Figure 3.9: Face images in (a) and (b) are captured when person 1 is at the positions shown as the red circles in (c) and (d) respectively

the person in the white circles as shown in Figure 3.9(d). For locations C and D, PTZ camera *C5* is selected. Person 1 is walking towards *C5*, but she isn't looking at her walking direction since she is avoiding another person. Hence the best frontal face images are not as good as expected. However, *C5* is the best choice out of the 4 PTZ cameras.

3.4 Conclusions

In this chapter, we present a scheduling scheme for a set of PTZ cameras to capture the best close-up frontal face images of a person of interest. The PTZ cameras are guided by the 3D head positions provided by the fixed overhead cameras. The best PTZ camera for each frame is selected based on the capture quality of frontal face image, which is evaluated by the obstruction level, position and velocity of the target and the parameter state of candidate PTZ cameras. When a new PTZ camera is waken up based on the capture quality, a handoff success probability, which describes the time that the new camera needed to arrive at the desired state from current state, is taken into account. The quantitative models are created for the two probabilities to help assigning the most suitable PTZ camera for the target. Experiments show that our method can capture high-quality close-up frontal face images effectively with most of them around the image centers. Our scheduling method can also be extended to capture close-up frontal face images of multiple targets by considering additional factors such as the capture priority.

4. VIRTUAL RECONSTRUCTION OF BROKEN CERAMIC VESSELS

This chapter presents a method to assist in the tedious process of reconstructing ceramic vessels from excavated fragments. The method exploits the fragments' color and geometric information coupled with a series of generic models constructed by the experts to produce a virtual reconstruction of what the original vessels may have looked like. Generic models are the 3D surface models generated based on the experts' historical knowledge of the period, provenance of the artifact, site locations, etc. The generic models need not to be identical to the original vessel, but must be within a geometric transformation of it in most parts. The surface markings on fragments and generic models are extracted based on the color information. By aligning a fragment against the corresponding generic model using the geometric relation between the markings on them, the ceramic vessels are virtually reconstructed. The alignment is based on a novel set of weighted discrete moments computed from convex hulls of the markings on the surface of the fragments and the generic vessels.

4.1 Introduction

Visualization and computer vision techniques have been applied in cultural heritage to facilitate the analysis and understandings of excavation findings, such as digital reconstruction of an ancient village [80], automatic 3D data acquisition [81-83] and documentation [84] of archaeological findings, identification of rotation axis of wheel-produced ceramics [84, 85], computing the profile sections

of fragments [86] and semi-automatically producing a collection of 3D vessels similar to those found in excavations [87]. In this chapter, we focus on virtually reconstructing/mending broken ceramic vessels.

Computational methods have been used to facility the image and document reassembly. Saharan and Singh [88] use the flood fill algorithm to obtain the closed boundaries of fragmented images and calculate the local curvature of each boundary pixel which is stored into a string. The fragment matching is then reduced to a string matching problem. Zhu et al. [89] propose to use turning-function-based partial curve matching to find the candidate matches and define the global consistency as the global criterion to do document reconstruction. The global match confidences are assigned to each candidate match and then these confidences are iteratively updated via the gradient projection method to maximize the criterion. Tsamoura and Pitas [90] instead present a color based approach to reassemble fragmented images and paintings. A neural network based color quantization approach for the representation of the image contour followed by a dynamic programming technique is employed to identify the matching contour segments of the image fragments. Aminogi et al. [91] utilize both the shape and the color characteristics of the image fragment contours. A contour pixel sequence is overlaid on another one and, for each such “placement”, the curvature and color differences of the corresponding contour pixels are estimated. If their total sum is less than a predefined threshold, the contour segments are considered to match.

Unlike images and documents, the reconstructions of solid artifact fragments are usually conducted in 3D space [92]. There are a variety of existing computer-

aided techniques for the reassembly of those fragments. Some works [93, 94] assume original models are available for digitization or scanning before reconstruction. By using the adaptive clustering and Self-Organizing Feature Map (SOFM) technique [95], Igwe and Knopf [93] establish the correspondences between the fragments and the original model. The transformations are estimated from SOFM and used to morph all fragments back to the original model. Thomas et al. [94] reconstruct a tibia by aligning each fragment's surface to the intact template using Geomagic Studio's built-in iterative registration function followed by an iterative closest point algorithm. The whole process depends on expert interaction. Unfortunately, in more general mending scenarios as in this paper, exact original models are not available, making the reassembling fragments more of a 3D puzzle-solving problem than aligning a shard to a vessel. Similar to the jigsaw puzzles in 2D, fracture surface information of 3D fragments is used to do the reconstruction. Huang et al. [14] compute a patch based surface feature clusters for all fracture surfaces and use the corresponding features to match all fracture surfaces pairwise. Without relying on any surface features, Winkelbach and Wahl [15] calculate the surface normal of the points on the fracture surface and declare two points in tangential contact if their normal directions are opposite. Two fragments are mended by changing the pose and position of one fragment until maximal fracture surface (point) contact is achieved. To efficiently compute fragment matches and avoid small erosion, Brown et al. [96] regularly resample fragment edges into a "ribbon" from which surface normals are calculated based on rearranged ribbon points.

The above reconstruction methods are based on the fracture surface, which may not be applied for thin-shell objects since the information on their fracture surfaces is limited. Additionally, the fracture surface may be unavailable or expensive to obtain under certain circumstances. To deal with this lack of information, Sağıroğlu and Erçil [97, 98] utilize textures along the fracture. They use inpainting and texture synthesis methods to predict the textures of a band outside the border of fragments, and then corresponding pieces and the transformations between them are found using Euclidian distances [97] or FFT based correlation [98] of the texture features on the predicted regions. The experiments are performed only on 2D fragments although the authors state that the method can be extended to 3D cases. In the absence of textures on the fragment borders, both methods will fail, but ours would work in that case, as it considers more than one feature. Some other works [16-19] first extract fracture contours and then find the matching fragments by curve matching. Instead of only using one kind of information of fragments, Papaioannou and Karabassi [99] combine curve matching and fracture surface matching techniques, hence allowing for the reassembly of both thin and thick shell objects. Similarly, Oxholm and Nishino [100] leverage both the geometric and color properties of fractures contour to establish the matching fragments.

No matter how many different types of information or tools are used, the above methods are fracture information based, relying on the geometry, color or texture of fracture contours and the geometry of fracture surfaces. Son et al. [101], however, use more than the fracture information. They estimate an axis of symmetry for each

fragment. When the estimated axes are reliable, fragment mending is based on both the symmetry property and break curve, otherwise is based solely on break curves.

The edges of archaeological fragments are vulnerable and can be eroded through time while in the ground or during excavation, as shown in Figure 4.1. This leaves the fracture information not well preserved hence making fragments difficult to be matched. Surface (not fracture surface) information of fragments, such as surface markings, on the other hand, is better preserved and the fragment surfaces carry valuable information useful for representing and reassembling fragments.

Figure 4.1: Ceramic fragments with eroded edges

Figure 4.2 shows the some fragments excavated from the Independence National Historical Park (INHP) in Philadelphia, Pennsylvania. They are thin-shell pieces dating from the late 17th to the mid-20th century. They are scanned at the INHP lab using a 3D scanner. Many of them have surface markings, which are the color patterns drawn on their surfaces.

Figure 4.2: Ceramic fragments

In this chapter, we propose to extract the surface markings on the fragments and generic models based on colors and then reconstruct a vessel by aligning fragments against the vessel′s generic models using the geometric cues of the corresponding markings on the fragments and generic model. The procedure is shown in Figure 4.3. For each extracted surface marking, we create a convex hull, resulting in a set of convex hulls on the fragment and the generic model. We introduce weighted moments to find the transformation that relates two 3D data sets (convex hull vertices) without the need of establishing point-to-point correspondences. Most of the previous works use moments of high order (up to fifth order in [102]) and handle up to similarity transformations between two data sets [103, 104]. We derive a novel weighted affine invariant moment with low order (zero and first order) to uniquely solve for affine transformation parameters. Candidate corresponding markings are established using absolute affine invariants derived from the weighted moments, which are computed from the convex hulls

associated with the set of markings. True marking correspondences are declared after a validation process. The local affine transformations are recovered from these true corresponding markings and used to transform and align the fragment against generic models. The average distance between the 3D points on the transformed shard and their closest points on the generic model is computed to evaluate the goodness of the alignment. This error is important as it confirms the best alignment or rejects it even when correspondences between markings on a fragment and a generic model are declared, or when more than one generic model is found to match a given fragment based on markings (in this case the generic model that results into the smallest average distance will be determined as the matching vessel).

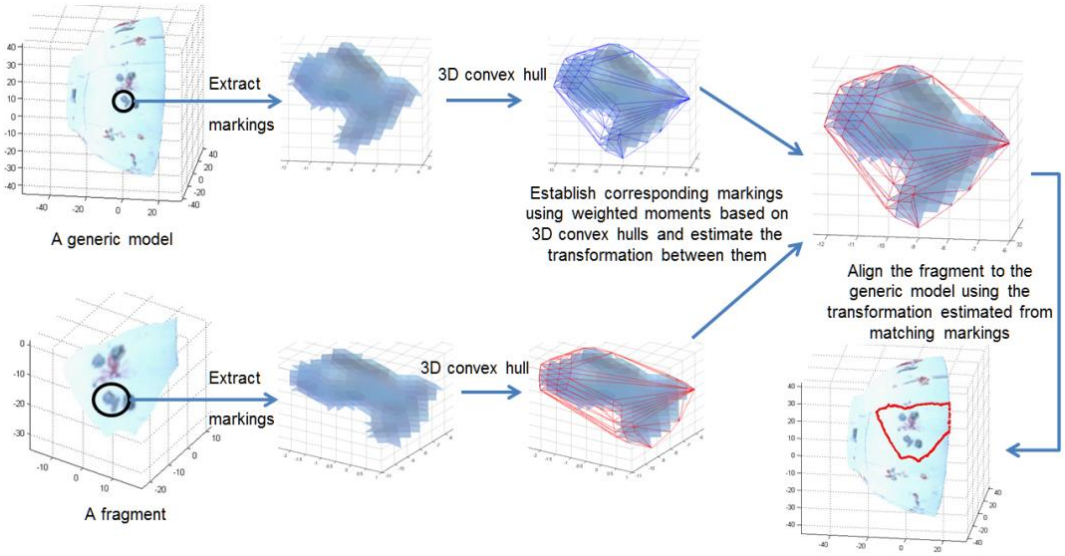


Figure 4.3: Aligning a fragment to a corresponding generic model based on color markings

4.2 Generic Models

Archeologists usually possess a library that consists of excavated broken/unbroken/mended artifacts, patterns of relief, color markings, and historical documents describing the shape and dimension of various artifacts. They come from various digs and from various eras. The creation of a set of generic models is a process that involves the archaeological experts (e.g., archaeologists from INHP in Philadelphia - collaborators on our NSF grant) who are assisted by graphics and computational engineers for rendering in 3D what the experts perceive and consider as possible generic representations for possible vessels in a given dig. A raw template can be one of those unbroken or manually mended vessels. It is scanned and imported into any 3D sculpture software such as ZBrush and Meshlab. When the unbroken/mended vessels are not available, we simply create a 3D template vessel in ZBrush with the information supplied by the archaeologists (e.g., height, neck size, belly size of a vase), and interactively modify the template in accordance with what they think are good generic models. The archeologists have also the option of applying different relief patterns on the template and making some local deformations if necessary. Normally, from one template, archeologists will generate dozens of variations which belong to one given category of vessels. Any substantial change in the template will result in a new category. The creation of the generic models is also a dynamic process. If the generic models in one category have very few fragments aligned to them, the archeologists will make changes to the shape, the relief patterns, the color markings or the location of these patterns to create new variations. In some cases, we have to create a new template (category)

based on newly excavated fragments. Finally, the process of automating the fragment mending based on the experts' opinion, even when that opinion does not reflect all the vessels in a given dig, results in a substantial reduction in the total fragments that the archeologists have to go through to manually mend them. The archeologists arrive at new historical findings by studying those sets for which they do not have good models. At a minimum, this proposed computational technology helps in pruning and reducing the data to a manageable set, freeing the archeologists from a laborious task such as mending what they know already. By and large, this is one of the main contributions that our work and similar work in computational archaeology offer.

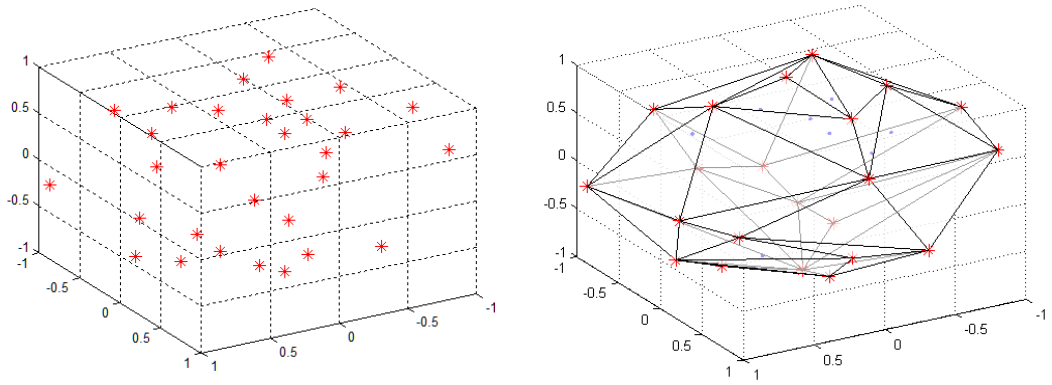
4.3 Modeling Fragments and Generic Models based on their Color Markings

The surface markings are the patterns drawn on a vessel and usually have different colors from most other parts of a vessel. The determination of surface markings needs experts' opinions and generic vessel models. For virtual reconstruction, either the whole surface markings or parts of the surface markings with certain colors are extracted. The color distribution of surface markings and background differs for different vessels thus the thresholds for surface marking extraction vary accordingly. In this work the surface markings on the generic models and fragments are extracted by manually thresholding the color information of the markings and/or the background. This can also be done by having an expert manually delineate the various surface markings.

For each surface marking, a 3D convex hull is computed based on the points of the marking using the algorithm introduced by Barber et al. [105]. The convex hull models a marking and is used towards aligning the individual ceramic fragment to the generic vessel. The convex hull is given by

$$Conv(M) = \left\{ \sum_{i=1}^{k} \lambda_i \boldsymbol{q_i} \,|\, \boldsymbol{q_i} \epsilon M, \lambda_i \geq 0, \forall i \epsilon \{1, \dots, k\}, \sum_{i=1}^{k} \lambda_i = 1 \,, k \epsilon \mathbb{N} \right\} \tag{4.1}$$

where M is the set of points on a marking and $\boldsymbol{q_i}$ is the *i-th* point in the set. For a set of points in 3D space, the convex hull is the smallest convex bounding polygon containing all the points, as shown in Figure 4.4 where the vertices and inner points of the convex hull are illustrated.



(a) The 3D data points (b) The convex hull of the 3D data points

Figure 4.4: A 3D point set and its convex hull where the red asterisks denote the vertices and the blue dots denote the rest of the points inside the convex hull

There are many reasons for using the convex hull as a marking representation[106]:

(i) The convex hull for a marking is unique.

(ii) It is a compact representation for a marking.

(iii) It is computationally efficient. The upper bound of the computational complexity for finding the convex hull of *n* data points is of order O(n log n).

(iv) It is affine invariant, which means that the convex hull of a data set subjected to an affine transformation is simply the affine transformed convex hull of the data before the transformation.

Markings that are different in shape (interior to the convex hull) may have identical convex hulls, and that would be a problem if we were to exclusively use the convex hull affine invariants to do the marking matching and fragment alignment. The convex hull affine invariant features are only used for establishing the candidate corresponding markings on a fragment and generic model. The true matching markings are declared after a validation process using all points on the makings not only the vertices of the convex hull of the markings (shown in section 4.5). This rules out the possibility of matching a fragment with a generic model grounded on different markings that have identical convex hulls.

4.4 Weighted Moments

Given a data set $R = \{\boldsymbol{v_i} = (x_i, y_i, z_i) | i = 1,2, \dots, n\}$, a density function $f(x_i, y_i, z_i)$ and an *s*-weighted function $w(\boldsymbol{v_i}, \boldsymbol{v_j}, \boldsymbol{v_k})_s$, we define the *s*-weighted central moment as

$$\mu(a,b,c)_s = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}(x_i - \bar{x})^a (y_i - \bar{y})^b (z_i - \bar{z})^c f(x_i, y_i, z_i) w(\boldsymbol{v_i}, \boldsymbol{v_j}, \boldsymbol{v_k})_s \tag{4.2}$$

where $(\bar{x}, \bar{y}, \bar{z})$ is the center of the data set. The choice of the weight function is dictated by a geometric transformation, and is based on the volume relative invariance of an affine transformation (section 4.5.1). Without loss of generality, we let $f(x_i, y_i, z_i) = 1$, hence equation (4.2) becomes

$$\mu(a,b,c)_s = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}(x_i - \bar{x})^a (y_i - \bar{y})^b (z_i - \bar{z})^c w(\boldsymbol{v_i}, \boldsymbol{v_j}, \boldsymbol{v_k})_s \tag{4.3}$$

4.5 Establishing Matching Markings based on their Geometric Relation

The corresponding markings on a fragment and a generic models are related by a geometric transformation: affine transformation. In this section, we establish a pair of matching markings by creating an absolute affine invariant based on the weighted moments. So an appropriate *s*-weighted function is first determined.

4.5.1 Choice of the S-weighted Function

The convex hull is used as a compressed representation of a surface marking, and the correspondences of the surface markings between fragments and generic models are established using the vertices of their convex hulls. Since the number of convex hull vertices is generally much less than the number of points of its corresponding surface marking, the computation load of the weighted moments is reduced using the convex hull representation. Let $\boldsymbol{v_i} = (x_i, y_i, z_i)$, $\boldsymbol{v_j} = (x_j, y_j, z_j)$ and $\boldsymbol{v_k} = (x_k, y_k, z_k)$ be three vertices on the convex hull of a marking, and let $\boldsymbol{v_{ai}} = (x_{ai}, y_{ai}, z_{ai})$, $\boldsymbol{v_{aj}} = (x_{aj}, y_{aj}, z_{aj}), \boldsymbol{v_{ak}} = (x_{ak}, y_{ak}, z_{ak})$ be their counterparts after an affine transformation $T_A = \{[L], \boldsymbol{B}\}$ with

$$[L] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{4.4}$$

$$\begin{bmatrix} x_{ai} & x_{aj} & x_{ak} \\ y_{ai} & y_{aj} & y_{ak} \\ z_{ai} & z_{aj} & z_{ak} \end{bmatrix} = [L] \begin{bmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \\ z_i & z_j & z_k \end{bmatrix} + \boldsymbol{B} \tag{4.5}$$

where $i, j, k = 1,2, \dots, n$, and where n is the number of convex hull vertices. Note that the centroids of the convex hull vertices also follow the affine transformation, i.e.

$$\begin{bmatrix} \bar{x}_a \\ \bar{y}_a \\ \bar{z}_a \end{bmatrix} = [L] \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} + \boldsymbol{B} \tag{4.6}$$

where $\boldsymbol{v_0} = (\bar{x}, \bar{y}, \bar{z})$ and $\boldsymbol{v_{a0}} = (\bar{x}_a, \bar{y}_a, \bar{z}_a)$ are the centroids of convex hull vertices of the original marking and its affine map, and $\boldsymbol{v_{a0}}^T$ and $\boldsymbol{v_0}^T$ are the transpose of $\boldsymbol{v_{a0}}$ and $\boldsymbol{v_0}$. Using equations (4.5) and (4.6), we have the following equation:

$$det\{[C_a]\} = det\{[L]\}det\{[C]\} \tag{4.7}$$

with $det\{[\,]\}$ being the determinant of the matrix [ ], and

$$C_a = \begin{bmatrix} x_{ai} - \bar{x}_a & x_{aj} - \bar{x}_a & x_{ak} - \bar{x}_a \\ y_{ai} - \bar{y}_a & y_{aj} - \bar{y}_a & y_{ak} - \bar{y}_a \\ z_{ai} - \bar{z}_a & z_{aj} - \bar{z}_a & z_{ak} - \bar{z}_a \end{bmatrix} \tag{4.8}$$

$$C = \begin{bmatrix} x_i - \bar{x} & x_j - \bar{x} & x_k - \bar{x} \\ y_i - \bar{y} & y_j - \bar{y} & y_k - \bar{y} \\ z_i - \bar{z} & z_j - \bar{z} & z_k - \bar{z} \end{bmatrix} \tag{4.9}$$

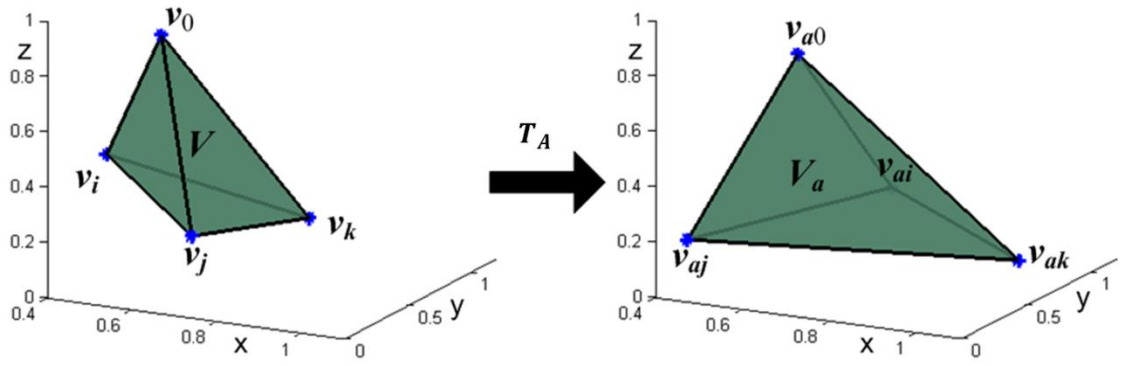


Figure 4.5: Geometric interpretation of equation (4.7). On the left there are four 3D points forming a tetrahedron, and on the right there are the corresponding points and tetrahedron after an affine transformation T_A.

Equation (4.7) has an interesting geometric interpretation. $\boldsymbol{v_i}$, $\boldsymbol{v_j}$ and $\boldsymbol{v_k}$ are three vertices on the convex hull and $\boldsymbol{v_0}$ is the centroid of all the convex hull vertices, hence they are not coplanar and form a tetrahedron as shown in left part of Figure 4.5. The corresponding affinely transformed convex hull vertices $\boldsymbol{v_{ai}}$, $\boldsymbol{v_{aj}}$, $\boldsymbol{v_{ak}}$ and their centroid $\boldsymbol{v_{a0}}$ also form a tetrahedron, as shown in right part of Figure 4.5. Let V_a and V be the volumes of the two tetrahedrons, then we have

$$V_a = \frac{1}{6}\left|(\boldsymbol{v_{ai}} - \boldsymbol{v_{a0}}) \cdot \left[\left(\boldsymbol{v_{aj}} - \boldsymbol{v_{a0}}\right) \times (\boldsymbol{v_{ak}} - \boldsymbol{v_{a0}})\right]\right| = \frac{1}{6}|det\{[C_a]\}| \tag{4.10}$$

$$V = \frac{1}{6}\left|(\boldsymbol{v_i} - \boldsymbol{v_0}) \cdot \left[\left(\boldsymbol{v_j} - \boldsymbol{v_0}\right) \times (\boldsymbol{v_k} - \boldsymbol{v_0})\right]\right| = \frac{1}{6}|det\{[C]\}| \tag{4.11}$$

From equations (4.7), (4.10) and (4.11), we can see as expected that the volume is preserved and is a relative invariant under the affine transformation T_A, with

$$V_a = |det\{[L]\}|V \tag{4.12}$$

With the volume being a relative affine invariant, we use it in defining the *s*-weighted affine invariant function in (4.2) and (4.3) to render it a relative affine invariant. The *s*-weighted function is given as

$$w(\boldsymbol{v_i}, \boldsymbol{v_j}, \boldsymbol{v_k})_s = |(\boldsymbol{v_i} - \boldsymbol{v_0}) \cdot [\left(\boldsymbol{v_j} - \boldsymbol{v_0}\right) \times (\boldsymbol{v_k} - \boldsymbol{v_0})]|^s \tag{4.13}$$

4.5.2 Absolute Invariants for Marking Matching

From equations (4.12) and (4.13), we can see that the weight function is linearly decomposed under the affine map

$$w_a(\boldsymbol{v_{ai}}, \boldsymbol{v_{aj}}, \boldsymbol{v_{ak}})_s = |det\{[L]\}|^s w(\boldsymbol{v_i}, \boldsymbol{v_j}, \boldsymbol{v_k})_s \tag{4.14}$$

,and so is the zero-*th* order *s*-weighted affine invariant central moments (using equations (4.3) and (4.14))

$$\mu_a(0,0,0)_s = |det\{[L]\}|^s \mu(0,0,0)_s \tag{4.15}$$

where

$$\mu(0,0,0)_s = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} w(\boldsymbol{v_i}, \boldsymbol{v_j}, \boldsymbol{v_k})_s \tag{4.16}$$

$$\mu_a(0,0,0)_s = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} w_a(\boldsymbol{v_{ai}}, \boldsymbol{v_{aj}}, \boldsymbol{v_{ak}})_s \tag{4.17}$$

Based on (4.15), absolute affine invariants can be constructed from these relative affine invariants by using any two different weight factors s_0 and s_1. We have

$$\mu_a(0,0,0)_{s_0} = |det\{[L]\}|^{s_0}\mu(0,0,0)_{s_0} \tag{4.18}$$

$$\mu_a(0,0,0)_{s_1} = |det\{[L]\}|^{s_1}\mu(0,0,0)_{s_1} \tag{4.19}$$

From (4.18) and (4.19), the following absolute affine invariant relation is obtained:

$$AI(s_0, s_1) = \frac{\sqrt[s_1]{\mu_a(0,0,0)_{s_1}}}{\sqrt[s_0]{\mu_a(0,0,0)_{s_0}}} = \frac{\sqrt[s_1]{\mu(0,0,0)_{s_1}}}{\sqrt[s_0]{\mu(0,0,0)_{s_0}}} \tag{4.20}$$

where $\mu_a(0,0,0)_{s_0}, \mu(0,0,0)_{s_0} \neq 0$ since $w_a(\boldsymbol{r_{ai}}, \boldsymbol{r_{aj}}, \boldsymbol{r_{ak}})_s, w(\boldsymbol{r_i}, \boldsymbol{r_j}, \boldsymbol{r_k})_s > 0$. The absolute invariant in (4.20) and a host of similar absolute invariants are used to establish which convex hull of a marking on the generic models corresponds to a given convex hull of the marking on the fragment. Corresponding convex hulls (not markings) are declared if the following inequality holds for a pre-determined threshold α (e.g., 0.05).

$$\frac{|AI_F - AI_G|}{AI_G} < \alpha \tag{4.21}$$

where AI_F is the absolute invariant of the convex hull of a marking on the fragment and AI_G the counterpart on the generic model. Usually, a relatively low α is picked

to avoid a lot of candidate corresponding markings. Once the convex hull correspondences are established, the best affine transformation that maps the convex hull on the fragment onto its corresponding one on the vessel is computed as explained in the next section. As mentioned in section 4.3, different markings might have the same convex hull, thus (4.21) can only establish the candidate matching markings on fragments and generic models. The true matching markings are determined not just based on the convex hull vertices (of the markings) but on the entire marking. This is done by transforming the marking points on a fragment to the generic model's domain using the transformation obtained in section 4.6 and then calculating the average distance error between all the marking points on the fragment and the generic model. The average distance error for a candidate matching marking pair with m and m' points on them respectively is

$$e_{MM'} = \frac{\sum_{k=1}^{m'} \min_{i=\{1,2,\dots,m\}} \|M'_k - M_i\|}{m'} \tag{4.22}$$

where M'_k denotes the $k\text{-}th$ point of the transformed marking M' on the fragment and M_i the $i\text{-}th$ point of the marking Mon the generic vessel. Usually, if the error is smaller than the scanner's resolution, the markings on the fragment and vessel are declared as true corresponding markings. The scanner's resolution can be determined as follows if it's not provided by the manufacture. Let $Fragment_i$ and $Fragment_j$ be the $i\text{-}th$ and $j\text{-}th$ point on a scanned fragment with n points, and the scanner's resolution is defined as

$$Res = \frac{\sum_{j=1}^{n} \min_{i=\{1,2,\dots,n\}, i\neq j} \left\|Fragment_i - Fragment_j\right\|}{n} \tag{4.23}$$

4.6 Estimating the Geometric Transformation between the Matching Markings

Given corresponding convex hull pairs, which have been determined using (4.21), we proceed to estimate the unique affine transformation $T_A = \{[L], \boldsymbol{B}\}$ that will align them. Towards that end, we use a set of first order s-weighted affine invariant central moments in (4.3) given by

$$\mu_a(1,0,0)_s = |\, det\{[L]\}\,|^s [a_{11}\ a_{12}\ a_{13}] \cdot [\mu(1,0,0)_s\ \ \mu(0,1,0)_s\ \ \mu(0,0,1)_s]^T \tag{4.24}$$

$$\mu_a(0,1,0)_s = |\, det\{[L]\}\,|^s [a_{21}\ a_{22}\ a_{23}] \cdot [\mu(1,0,0)_s\ \ \mu(0,1,0)_s\ \ \mu(0,0,1)_s]^T \tag{4.25}$$

$$\mu_a(0,0,1)_s = |\, det\{[L]\}\,|^s [a_{31}\ a_{32}\ a_{33}] \cdot [\mu(1,0,0)_s\ \ \mu(0,1,0)_s\ \ \mu(0,0,1)_s]^T \tag{4.26}$$

where $[\ ]^T$ is the transpose of matrix $[\ ]$. To solve the transformation $[L]$ (shown in equation (4.4)) that has 9 unknowns, a set of 3 different s values (the selection of the weight factor s follows the same choice as in [107]) is used in equation set (4.24), (4.25), (4.26) to arrive at 9 equations with 9 unknowns. As $|\, det\{[L]\}\,|^s$ is unknown, we need to eliminate it. This is achieved by dividing the equation set (4.24), (4.25), (4.26) by equation (4.15) for the 3 different s values. This results in the linear equation

$$[Mu_a] = [L][Mu] \tag{4.27}$$

with

$$[Mu] = \begin{bmatrix} \frac{\mu(1,0,0)_{s1}}{\mu(0,0,0)_{s1}} & \frac{\mu(1,0,0)_{s2}}{\mu(0,0,0)_{s2}} & \frac{\mu(1,0,0)_{s3}}{\mu(0,0,0)_{s3}} \\ \frac{\mu(0,1,0)_{s1}}{\mu(0,0,0)_{s1}} & \frac{\mu(0,1,0)_{s2}}{\mu(0,0,0)_{s2}} & \frac{\mu(0,1,0)_{s3}}{\mu(0,0,0)_{s3}} \\ \frac{\mu(0,0,1)_{s1}}{\mu(0,0,0)_{s1}} & \frac{\mu(0,0,1)_{s2}}{\mu(0,0,0)_{s2}} & \frac{\mu(0,0,1)_{s3}}{\mu(0,0,0)_{s3}} \end{bmatrix} \quad (4.28)$$

$$[Mu_a] = \begin{bmatrix} \frac{\mu_a(1,0,0)_{s1}}{\mu_a(0,0,0)_{s1}} & \frac{\mu_a(1,0,0)_{s2}}{\mu_a(0,0,0)_{s2}} & \frac{\mu_a(1,0,0)_{s3}}{\mu_a(0,0,0)_{s3}} \\ \frac{\mu_a(0,1,0)_{s1}}{\mu_a(0,0,0)_{s1}} & \frac{\mu_a(0,1,0)_{s2}}{\mu_a(0,0,0)_{s2}} & \frac{\mu_a(0,1,0)_{s3}}{\mu_a(0,0,0)_{s3}} \\ \frac{\mu_a(0,0,1)_{s1}}{\mu_a(0,0,0)_{s1}} & \frac{\mu_a(0,0,1)_{s2}}{\mu_a(0,0,0)_{s2}} & \frac{\mu_a(0,0,1)_{s3}}{\mu_a(0,0,0)_{s3}} \end{bmatrix} \quad (4.29)$$

The transformation $[L]$ is uniquely computed from (4.27). Once $[L]$ is found, the translation parameters $\boldsymbol{B}$ can be obtained from equation (4.6).

4.7 Aligning Fragments against Generic Models

With the corresponding markings and the estimated transformations, the correspondence between fragments and generic models is to be determined. The matching of a fragment to one of many generic models should not be solely based on the matching markings on the fragment and the generic model since marking points are just a subset of the data on the fragment and not the entire 3D surface data of the fragment. Hence it is necessary and imperative to use the average distance error between all the 3D points (not only marking points) on the transformed fragment and their closest points on the vessel to evaluate the goodness of the alignment. The transforming of the fragment into the vessel coordinate space is done in accordance with the transformation estimated (as in the previous section) from the markings on the fragment and its corresponding one on the vessel. The

alignment error is computed in terms of the scanner's resolution Res (shown in (4.23)), and is given as

$$e_{align} = \frac{\sum_{k=1}^{p} \min_{j=\{1,2,\ldots,n\}} \left\| Fragment_k - Vessel_j \right\|}{p \cdot Res} \tag{4.30}$$

where $Fragment_k$ is the $k\text{-}th$ point of the transformed fragment that contains p points and $Vessel_j$ the $j\text{-}th$ point of the generic model that contains n points. This error is useful to: (a) decide whether or not a fragment should be aligned to a given vessel after the markings on the fragment and the vessel are found to correspond by using equation (4.22). This is done by requiring the average distance error to be less than a predetermined threshold usually less than the scanner resolution; (b) select the best alignment when several markings on a fragment have their own respective true corresponding ones on a generic model. Each corresponding marking pair can generate a transformation and an alignment of the fragment to the generic model. Of all the possible alignments, the one with minimal alignment error is selected as the best one; and (c) disambiguate the case where there is more than one generic vessel that has markings that correspond to the ones on the fragment, by choosing the vessel for which that distance error is minimal.

4.8 Experiments

4.8.1 Data Collection

The ceramic artifact collection dug from the National Constitution Center site in Independence Park is used as a data set in this paper. The artifact collection is currently kept at Independence Living History Center in Philadelphia, PA. Some of

the artifacts are mended by experts using glue or bands. Suitable samples of that artifact collection are chosen as to serve as important bases of generic models and test samples. Fragments are scanned at the INHP lab using a Konica Minolta Vivid 910 3D scanner. The Konica scanner employs laser-beam light sectioning technology to scan work pieces using a slit beam. Light reflected from the work piece is acquired by a CCD camera, and 3D data is then created by triangulation to determine distance information. The scan setup and scanning equipment are shown in Figure 4.6. The black carpet on the floor is to create a clean background so that the scanned fragments do not blend in.

The scanned raw data (discrete 3D points) is then transferred to the software Geomagic Studio via Konica designed plugins. Geomagic Studio performs a version of Delaunay triangulation producing a triangulation representation or triangle mesh of the discrete 3D point set.

Figure 4.6: 3D scanner and scan setup

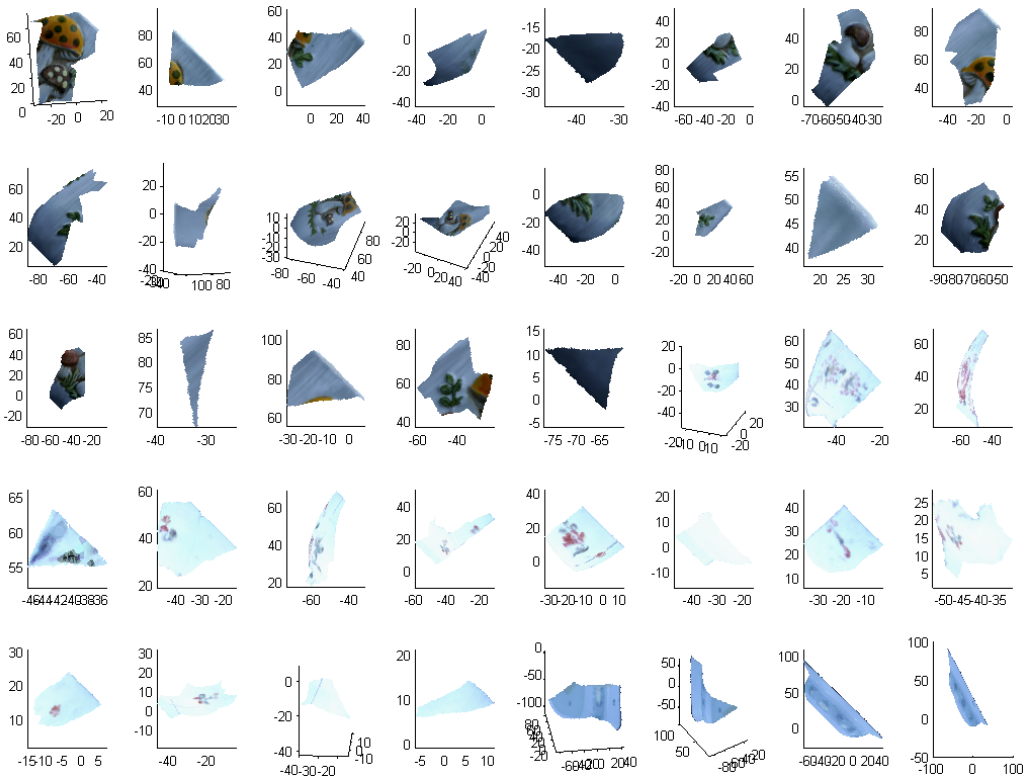
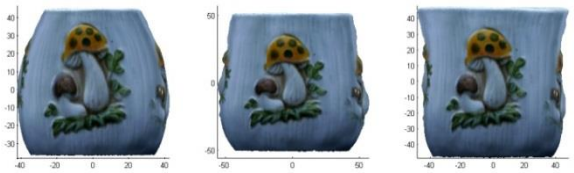


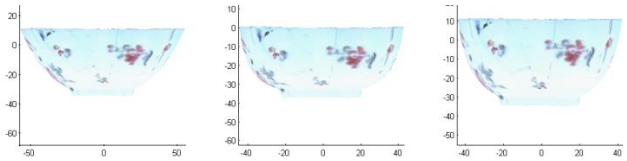
Figure 4.7: 3D scanned fragments from the vessels with marking on them

4.8.2 Alignment Results

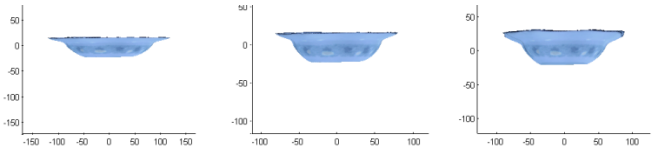
We test our methods on 174 fragments that are from the vessels having markings and part of them are shown in Figure 4.7. According to archaeologists′ observations, the 174 fragments are from 5 types of vessels: cup, bowl, plate, vase, and jar. They provide 20+ generic models for each type of vessel, hence a total of 100+ possible generic models. We show 3 generic models for each category (cup, bowl, and plate) in Figure 4.8. The 174 fragments are mixed when we do the virtual mending.



(a) 3 generic models for cups



(b) 3 generic models for bowls



(c) 3 generic models for plates

Figure 4.8: Some of generic models for different types of vessels

The reconstruction results of the 174 fragments are shown in Figure 4.9 and Table 4.1. The number of fragments for each vessel is determined after the experts finished mending all the fragments by hand. We find that most fragments from a vessel with markings are aligned against one generic model in each vessel category. The fragments bounded by red boundaries in Figure 4.9 are mended using the markings on the fragments and the generic models. We also find that a small number of fragments are not aligned to the correct generic model based on markings. These fragments, highlighted using green boundaries in Figure 4.9, are aligned to the corresponding generic model manually to show what the original vessels look like and illustrate our reconstruction result better. There are several reasons for the

result that not all the fragments are aligned: 1) the best generic model (the generic model that is closest to the original vessel in terms of shapes and colors) is not exact enough; or 2) the markings on the fragment are occluded; or 3) there is no color marking on the fragment although it is from a vessel with markings, i.e., markings reside on other fragments of the vessel, as shown in Figure 4.10 where the two fragments, from the vessels in Figure 4.9(e) and (g) respectively, have no marking on them. The remaining unrecovered parts, shown as the white 'holes' in Figure 4.9, are the missing fragments which are usually small.

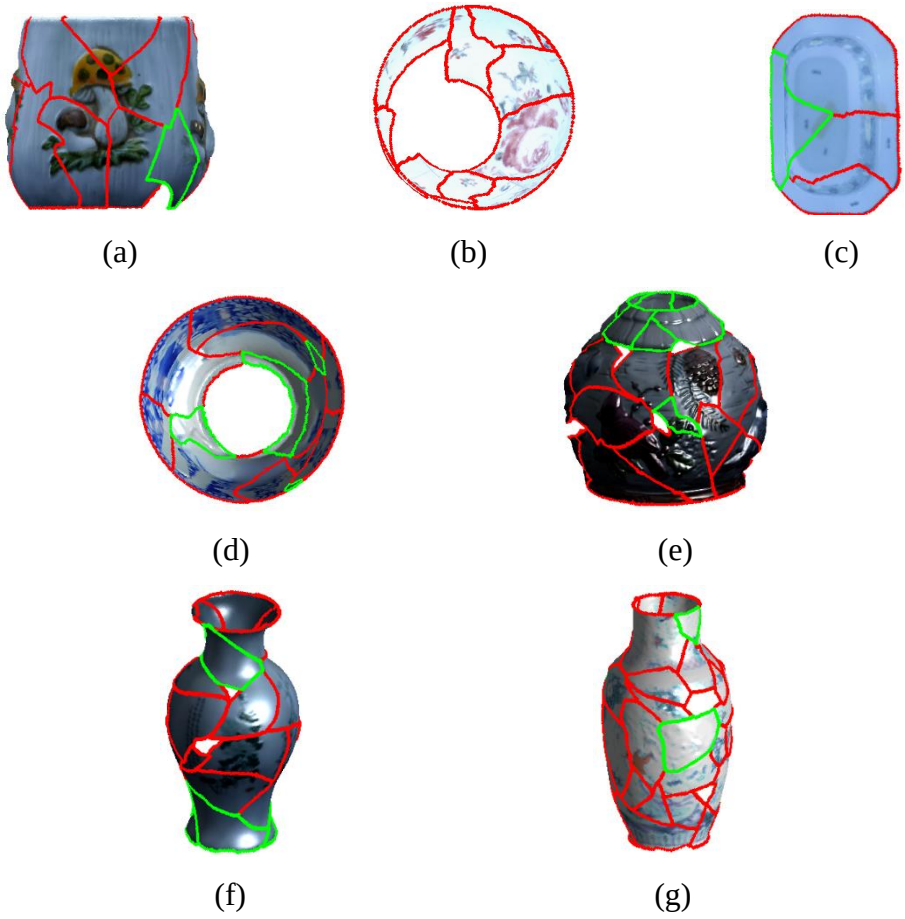


Figure 4.9: Alignment results where fragments with red boundaries are aligned using our method and those with green boundaries are aligned manually

(a) A fragment from Figure 4.9(d) (b) A fragment from Figure 4.9(f)

Figure 4.10: Fragments without any markings on them

Table 4.1: Reconstruction results and errors

Vessel ID	# of fragments from each vessel	# of correctly mended fragments	# of not aligned fragments	# of misaligned fragments	Normalized reconstruction error
(a)	21	17	4	1	0.64
(b)	13	13	0	0	0.69
(c)	4	3	1	1	0.74
(d)	14	10	4	0	0.58
(e)	47	33	14	2	0.70
(f)	34	25	9	1	0.67
(g)	41	36	5	0	0.61

From Table 4.1, around 80% of fragments (137 out of 174 fragments) are aligned to the best generic models. The misaligned fragments are those aligned to other generic models instead of the best ones. Among the fragments that are not aligned to the best generic models (highlighted using green boundaries in Figure 4.9), a small portion (5 out of 37) are misaligned. For instance, 2 fragments from

Figure 4.9(a) and (c) are misaligned to other generic models, as shown in Figure 4.12. The reconstruction errors are normalized using the resolution of the 3D data and are smaller than the resolution, as shown in Table 4.1. If we look at the 3D reconstructed results, around 85% of surface areas of the vessels are recovered using our method, shown as the surface surrounded by the red boundaries in Figure 4.9. For the very small portion of the surface of a vessel that are not reconstructed, the experts can manage to mend them manually without too much effort, or different aspects of a fragment other than markings, such as parabolic contours [108] of on a fragment surface, can be exploited. This can deal with fragments and even a whole vessel without any color markings. Figure 4.11 shows 2 vessels of this kind.

Figure 4.11: Two misaligned fragments

Figure 4.12: Two vessels without any color markings

4.9 Conclusions

We present a method to reconstruct vessels virtually by utilizing both the color and geometric cues of surface markings on fragments. 3D scanned fragments are aligned against generic vessels based on surface markings. This is one of many tools that computational archaeology can exploit towards helping in the mending of archaeological artifacts. The method weighs between expert opinion (with expected uncertainties) and total lack of prior knowledge. Built-in uncertainties in the generic models allow for rotation, scaling, shifting, and shearing between markings on the generic models and their corresponding ones on the fragments.

The method is shown to be reliable on the sample set most of which is chosen from the INPH ceramic artifact collection recovered from the National Constitution Center site. The results are reported in terms of normalized residual error statistics and the reconstructed 3D surfaces, and the reported average errors are within the range of the 3D scanner's resolution. The work has focused on the use of one aspect (surface markings) amongst many embedded in the shards. This, in conjunction with many other aspects such as surface color, texture, or cracks, could be collectively used as enabling technology helping in the mending process for fragments with/without markings. The whole project as an application of computer vision in archaeology is unique for timely analysis, interpretation, and presentation of history evidence. It is also considered as a great need by the U.S. Department of the Interior National Park Service.

5. CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

In this thesis, a smart camera network consisting of two overhead cameras and a set of PTZ cameras are used to approximate an ‘ideal tracker’ which generates the 3D positions of pedestrians in crowded scenes and captures close-up frontal face image of a person(s) of interest (target(s)) in the image center across time and space. Overhead cameras track people in 3D by fusing extracted geometric and color cues. A potential head top segment for each person in the scene is detected mainly using projective geometry, from which the 3D head point is obtained efficiently without using the full disparity map that is usually used to do 3D localization. If the detected head point is not accurate enough, the head top color distribution over frames (mutual information) is used to correct the estimated head position across consecutive frames. With these two complementary information people are tracked accurately and robustly. For the crowded scene with 14 people inside an area of around 4 square meters, the tracking errors for ground positions and heights are around 4 cm and 3 cm, respectively.

The 3D position of a person generated by the static overhead cameras is used to guide a set of PTZ cameras to capture close-up frontal face images of a target across time. The best PTZ camera is selected in each frame for face image acquisition, based on capture quality and handoff success probabilities that take into account the degree and origin of occlusion, the constraints on camera movement and its physical parameters (pan, tilt and zoom), and the quality of the

frontal face image of the target at any time. Experiments show that the scheduling scheme is effective to capture high-quality close-up frontal face images.

The geometric and color cues of ceramic fragments are used to reconstruct broken vessels virtually. The experts' knowledge is also integrated by using the prior generic models created by them. A degree of uncertainty and variability is allowed in the building of the generic model. The expert model is learned through approximations to the excavated vessel or through knowledge of the historical era. The markings on the surface of fragments and generic vessel models are extracted based on colors, and then fragments are aligned to the corresponding generic models using the geometric relations between them. The approach is tested on some fragments from INHP, and the reconstruction errors are smaller than the 3D scanner's resolution.

5.2 Future Directions

In the thesis, pedestrians are tracked using geometric cues under common motion assumption using the mean shift algorithm based on color cues between frames to improve on the head position estimates over frames. Either the detected or the predicted head point is taken as a pedestrian's position. In the case of real videos, the uncertainties of detected head point can be larger, thus other schemes such as an extended Kalman filter [109] that weighs between the detected head position based on the current frame (i.e., the measurement) and the predicted value originating from the estimated motion derived from previous frames can be employed. Based on the positon in the previous frame and a transition model, the

pedestrian's position in the current frame would be determined. The transition model would consider the uncertainties of measurements and predicted values. One can also relax the assumption of constant velocity between frames to improve the tracking robustness and allow for abrupt motions (as abrupt stop or turn).

In the thesis, we extract the foreground using background subtraction in HSV color space. It works for virtual scenes, however, this method is apt to fail in the real world scenario where more background clutter exists. A mixture of Gaussians can be used to model the color of each pixel, yielding a robust segmentation of the foreground in spite of the background clutter. This would allow the extension of our tracking method to outdoor as well as indoor scenes.

When close-up face image capture is needed for multiple persons, additional factors not mentioned in this thesis are considered. For example, we would need to consider a capture priority factor, where more attention is given to people who would be the first to walk outside of the FOV; or in some other applications more attention would be given to pedestrians who just enter the FOV (high security areas, or "do not enter" areas, etc.). Hence, the relative positions of people in the scene compared with the FOV boundaries and the walking directions would be important factors to be considered. We also need to create a mechanism which recognizes when the task of capturing the face image of one target is achieved so that the occupying PTZ camera can be released and be available if needed in tracking the target of next highest priority. The ending of the capturing task will be triggered when an adequate number of high quality frontal face images are captured.

For the reconstruction of broken vessels, only the markings on the ceramic fragments and expert generic modes are considered in the thesis. This approach would then fail if there are no markings on the fragments or markings are severely eroded. There are many other aspects on fragments and vessels that one can exploit, such as break boundaries of fragments and intrinsic surface features (e.g., differential geometry features on fragment surface [108]). We should also integrate many aspects/tools to improve the virtual reconstruction result where more vessels with/without markings are recovered more accurate. As a start, we have used 2 aspects [110], surface markings and anchor points on break boundaries in vessel reconstruction. To utilize multiple aspects of fragments, we need to first investigate the features on each fragment (such as color markings, reliefs, textures and conditions of break boundaries) to figure out what aspects can be used and then decide on optimal ways of combining them.

LIST OF REFERENCES

[1] R. T. Collins, A. J. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," in *Proceedings of the IEEE*, 2001, pp. 1456-1477.

[2] H. Aghajan and A. Cavallaro, *Multi-camera networks: principles and applications*: Academic press, 2009.

[3] G. R. Bradski, "Computer Vision Face Tracking For Use in a Perceptual User Interface," in *Workshop on Applications of Computer Vision*, 1998, pp. 214-219.

[4] L. Tyng-Luh and C. Hwann-Tzong, "Real-time tracking using trust-region methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 397-402, 2004.

[5] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2000, pp. 142-149.

[6] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," presented at the European Conference on Computer Vision, 2002.

[7] T. H. Chang and G. Shaogang, "Tracking multiple people with a multi-camera system," in *IEEE Workshop on Multi-Object Tracking*, 2001, pp. 19-26.

[8] Q. Cai and J. K. Aggarwal, "Tracking human motion using multiple cameras," in *13th International Conference on Pattern Recognition*, 1996, pp. 68-72.

[9] S. Khan and M. Shah, "Consistent labeling of tracked objects in multiple cameras with overlapping fields of view," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1355-1360, 2003.

[10] M. Taj and A. Cavallaro, "Multi-camera track-before-detect," in *Third ACM/IEEE International Conference on Distributed Smart Cameras*, 2009, pp. 1-6.

[11] L. Lee, R. Romano, and G. Stein, "Monitoring activities from multiple video streams: establishing a common coordinate frame," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 758-767, 2000.

[12] A. Mittal and L. Davis, "Unified multi-camera detection and tracking using region-matching," in *IEEE Workshop on Multi-Object Tracking*, 2001, pp. 3-10.

[13] P. L. Jeppson. (2009, September 7, 2009). *New methods -- and forthcoming records!* *Http://digginginthearchives.Blogspot.Com*.

[14] Q. Huang, S. Flöry, N. Gelfand, M. Hofer, and H. Pottmann, "Reassembling fractured objects by geometric matching," *ACM Transactions on Graphics*, vol. 25, pp. 569-578, 2006.

[15] S. Winkelbach and F. M. Wahl, "Pairwise matching of 3d fragments using cluster trees," *International Journal of Computer Vision*, vol. 78, pp. 1-13, 2008.

[16] J. C. McBride and B. B. Kimia, "Archaeological fragment reconstruction using curve-matching," in *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW '03. Conference on*, 2003, pp. 3-10.

[17] G. Ucoluk and I. H. Toroslu, "Automatic reconstruction of broken 3-d surface objects," *Computers & Graphics*, vol. 23, pp. 573-582, Aug 1999.

[18] W. Rodriguez, M. Last, A. Kandel, and H. Bunke, "3-dimensional curve similarity using string matching," *Robotics and Autonomous Systems*, vol. 49, pp. 165-172, 2004.

[19] L. Y., H. Gardner, H. Jin, N. Liu, R. Hawkins, and I. Farrington, "Interactive reconstruction of archaeological fragments in a collaborative environment," in *Proceedings of the IEEE 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications*, 2007, pp. 23-29.

[20] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, pp. 509-522, 2002.

[21] A. Fusiello, "Elements of geometric computer vision," *Available from: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/FUSIELLO4/tutorial.html*, 2006.

[22] C. Nieuwenhuis and D. Cremers, "Spatially varying color distributions for interactive multilabel segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 1234-1247, 2013.

[23] T. Gevers and A. W. M. Smeulders, "Color-based object recognition," *Pattern Recognition*, vol. 32, pp. 453-464, 1999.

[24] M. J. Swain and D. H. Ballard, "Color indexing," *International journal of computer vision*, vol. 7, pp. 11-32, 1991.

[25] J.-S. Lee, "Digital image enhancement and noise filtering by use of local statistics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 165-168, 1980.

[26] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 679-698, 1986.

[27] J. Wang, C. Zhang, and H. Shum, "Face image resolution versus face recognition performance based on two global methods," in *Asia Conference on Computer Vision*, 2004.

[28] J. Daugman, "How iris recognition works," in *International Conference on Image Processing*, 2002, pp. 33-36.

[29] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "3d head tracking for fall detection using a single calibrated camera," *Image and Vision Computing*, vol. 31, pp. 246-254, 2013.

[30] T. Zhao and R. Nevatia, "Tracking multiple humans in complex situations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1208-1221, 2004.

[31] S. J. D. Prince, J. H. Elder, Y. Hou, and M. Sizinstev, "Pre-attentive face detection for foveated wide-field surveillance," in *IEEE Workshop on Applications on Computer Vision*, 2005, pp. 439-446.

[32] G. Brostow and R. Cipolla, "Unsupervised bayesian detection of independent motion in crowds," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 594-601.

[33] J. Orwell, S. Massey, P. Remagnino, D. Greenhill, and G. Jones, "A multi-agent framework for visual surveillance," in *IEEE International 1st Conference on Image Processing*, 1999.

[34] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Sha, "Multi-camera multi-person tracking for easy living," in *Third IEEE International Workshop on Visual Surveillance*, 2000.

[35] A. Mittal and S. Larry, "M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene," vol. 51, pp. 189-203, 2003.

[36] S. M. Khan and M. Shah, "A multi-view approach to tracking people in crowded scenes using a planar homography constraint," in *European Conference on Computer Vision*, 2006, pp. 133-146.

[37] L. Sun, H. Di, L. Tao, and G. Xu, "A robust approach for person localization in multi-camera environment," in *International Conference on Pattern Recognition*, 2010, pp. 4036-4039.

[38] Z. Zhang and F. Cohen, "Pedestrian tracking based on 3d head point detection," in *International Conference on Computer Vision Theory and Applications (2)*, 2013, pp. 382-385.

[39] Z. Zhang and F. Cohen, "3d pedestrian tracking based on overhead cameras," in *International Conference on Distributed Smart Cameras* 2013, pp. 1-6.

[40] R. Eshel and Y. Moses, "Homography based multiple camera detection and tracking of people in a dense crowd," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1-8.

[41] S. M. Khan and M. Shah, "Tracking multiple occluding people by localizing on multiple scene planes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 505-519, 2009.

[42] D. Delannay, N. Danhier, and C. D. Vleeschouwer, "Detection and recognition of sports(wo)man from multiple views," in *ACM/IEEE International Conference on Distributed Smart Cameras*, 2009, pp. 1-7.

[43] T. T. Santos and C. H. Morimoto, "Multiple camera people detection and tracking using support integration," *Pattern Recognition Letters*, vol. 32, pp. 47-55, 2011.

[44] O. Ozturk, T. Yamasaki, and K. Aizawa, "Tracking of humans and estimation of body/head orientation from top-view single camera for visual focus of attention analysis," in *International Conference on Computer Vision*, 2009, pp. 1020-1027.

[45] N. Bellotto, E. Sommerlade, B. Benfold, C. Bibby, I. Reid, D. Roth, *et al.*, "A distributed camera system for multi-resolution surveillance," in *ACM/IEEE International Conference on Distributed Smart Cameras*, 2009, pp. 1-8.

[46] M. Boltes, A. Seyfried, B. Steffen, and A. Schadschneider, "Automatic extraction of pedestrian trajectories from video recordings," in *Pedestrian and Evacuation Dynamics 2008*, W. W. F. Klingsch, C. Rogsch, A. Schadschneider, and M. Schreckenberg, Eds., ed, 2010, pp. 43-54.

[47] D. Beymer, "Person counting using stereo," in *Workshop on Human Motion*, 2000, pp. 127-133.

[48] T. V. Oosterhout, S. Bakkes, and B. J. A. Kröse, "Head detection in stereo data for people counting and segmentation," in *International Conference on Computer Vision Theory and Applications*, 2011, pp. 620-625.

[49] T. V. Oosterhout, B. J. A. Kröse, and G. Englebienne, "People counting with stereo cameras - two template-based solutions," in *International Conference on Computer Vision Theory and Applications (2)* 2012, pp. 404-408.

[50] M. Boltes and A. Seyfried, "Collecting pedestrian trajectories," *Neurocomputing*, vol. 100, pp. 127-133, 2013.

[51] E. Oto, F. Lau, and H. Aghajan, "Color-based multiple agent tracking for wireless image sensor networks," in *8th International Conference on Advanced Concepts For Intelligent Vision Systems*, Antwerp, Belgium, 2006, pp. 299-310.

[52] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 780-785, 1997.

[53] X. Gao, T. E. Boult, F. Coetzee, and V. Ramesh, "Error analysis of background adaption," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2000, pp. 503-510

[54] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 747-757, 2000.

[55] T. Horprasert, D. Harwood, and L. S. Davis, "A robust background subtraction and shadow detection," in *Asian Conference on Computer Vision*, 2000.

[56] L. Vincent, "Gray scale area openings and closings, their efficient implementation and applications," in *Workshop on Mathematical Morphology Applications Signal Processing*, 1993, pp. 22-27.

[57] G. R. Taylor, A. J. Chosak, and P. C. Brewer, "OVVV: using virtual worlds to design and evaluate surveillance systems," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1-8.

[58] F. Qureshi and D. Terzopoulos, "Smart camera networks in virtual reality," *Proceedings of the IEEE*, vol. 96, pp. 1640-1656, 2008.

[59] F. Z. Qureshi and D. Terzopoulos, "Surveillance camera scheduling: A virtual vision approach," *Multimedia Systems*, vol. 12, pp. 269-283, 2006.

[60] W. Shao and D. Terzopoulos, "Autonomous pedestrians," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2005, pp. 19-28.

[61] F. C. Crow, "Summed-area tables for texture mapping," in *SIGGRAPH*, 1984, pp. 207-212.

[62] O. Veksler, "Fast variable window for stereo correspondence using integral images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 556-561.

[63] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, pp. 137-154, 2004.

[64] P. Fieguth and D. Terzopoulos, "Color-based tracking of heads and other mobile objects at video frame rates," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 21-27.

[65] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1998, pp. 232-237.

[66] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 564-577, 2003.

[67] H. Schweitzer, J. Bell, and F. Wu, "Very fast template matching," in *Computer Vision—ECCV 2002*, ed: Springer, 2002, pp. 358-372.

[68] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image and Vision Computing*, vol. 21, pp. 99-110, 2003.

[69] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Scale and orientation adaptive mean shift tracking," *IET Computer Vision*, vol. 6, pp. 52-61, 2012.

[70] T. Kailath, "The divergence and bhattacharyya distance measures in signal selection," *IEEE Transactions on Communication Technology*, vol. 15, pp. 52-60, 1967.

[71] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 24, pp. 603-619, 2002.

[72] X. Zhou, R. T. Collins, T. Kanade, and P. Metes, "A master-slave system to acquire biometric imagery of humans at distance," in *First ACM SIGMM international workshop on Video surveillance*, 2003, pp. 113-120.

[73] S. T. Stillman, R. Tanawongsuwan, and I. A. Essa, "A System for Tracking and Recognizing Multiple People with Multiple Camera," in *Second International Conference on Audio-Visionbased Person Authentication*, 1998, pp. 96-101.

[74] L. Marchesotti, L. Marcenaro, and C. Regazzoni, "Dual camera system for face detection in unconstrained environments," in *International Conference on Image Processing*, 2003, pp. 681-684.

[75] C. J. Costello, C. P. Diehl, A. Banerjee, and H. Fisher, "Scheduling an active camera to observe people," in *The ACM 2nd International Workshop on Video Surveillance & Sensor Networks*, 2004, pp. 39-45.

[76] A. Hampapur, S. Pankanti, A. Senior, Y.-L. Tian, L. Brown, and R. Bolle, "Face cataloger: Multi-scale imaging for relating identity to location," in *IEEE Conference on Advanced Video and Signal Based Surveillance*, 2003, pp. 13-20.

[77] A. D. Bimbo and F. Pernici, "Towards on-line saccade planning for high-resolution image sensing," *Pattern Recognition Letters*, vol. 27, pp. 1826-1834, 2006.

[78] N. Krahnstoever, T. Yu, S.-N. Lim, K. Patwardhan, and P. Tu, "Collaborative real-time control of active cameras in large scale surveillance systems," in *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, 2008.

[79] F. Z. Qureshi and D. Terzopoulos, "Planning ahead for PTZ camera assignment and handoff," in *Third ACM/IEEE International Conference on Distributed Smart Cameras*, , 2009, pp. 1-8.

[80] S. Olsen, A. Brickman, and Y. Cai, "Discovery by reconstruction: Exploring digital archeology," in *SIGCHI Workshop (Ambient Intelligence for Scientific Discovery (AISD))*, Vienna, 2004.

[81] B. J. Brown and S. Rusinkiewicz, "Global non-rigid alignment of 3-d scans," in *ACM SIGGRAPH*, San Diego, California, 2007.

[82] A. Bujakiewicz, M. Kowalczyk, P. Podlasiak, and D. Zawieska, "3d reconstruction and modelling of the contact surfaces for the archaeological small museum pieces," in *Proceedings of the ISPRS Commission V Symposium 'Image Engineering and Vision Metrology'*, Dresden, Germany, 2006.

[83] L. V. Gool and R. Sablatnig, "Special issue on 3d acquisition technology for cultural heritage," *Machine Vision and Applications*, vol. 17, pp. 347-348, Dec 2006.

[84] H. Mara, M. Kampel, F. Niccolucci, and R. Sablatnig, "Ancient coins & ceramics – 3d and 2d documentation for preservation and retrieval of lost heritage," in *Proceedings of the 2nd ISPRS International Workshop 3D-ARCH*, Zurich, Switzerland, 2007.

[85] A. Karasik and U. Smilansky, "3d scanning technology as a standard archaeological tool for pottery analysis: Practice and theory," *Journal of Archaeological Science,* vol. 35, pp. 1148-1168, May 2008.

[86] M. Kampel and R. Sablating, "3d data retrieval of archaeological pottery," in *Lecture Notes in Computer Science*. vol. 4270, H. Zha, Z. Pan, H. Thwaites, A. C. Addison, and M. Forte, Eds., ed, 2006, pp. 387-395.

[87] A. Koutsoudis, G. Pavlidis, F. Arnaoutoglou, D. Tsiafakis, and C. Chamzas, "Qp: A tool for generating 3d models of ancient greek pottery," *Journal of Cultural Heritage,* vol. 10, pp. 281-295, June 2009.

[88] R. Saharan and C. V. Singh, "Reassembly of 2d fragments in image reconstruction," *International Journal of computer applications,* vol. 19, pp. 41-45, 2011.

[89] L. Zhu, Z. Zhou, and D. Hu, "Globally consistent reconstruction of ripped-up documents," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 30, pp. 1-13, 2008.

[90] E. Tsamoura and I. Pitas, "Automatic color based reassembly of fragmented images and paintings," *IEEE Transactions on Image Processing,* vol. 19, pp. 680-690, 2010.

[91] F. Amigoni, S. Gazzani, and S. Podico, "A method for reassembling fragments in image reconstruction," in *International Conference on Image Processing*, 2003, pp. 581-584.

[92] F. Kleber and R. Sablatnig, "A survey of techniques for document and archaeology artefact reconstruction," in *International Conference on Document Analysis and Recognition* 2009, pp. 1061-1065.

[93] P. C. Igwe and G. K. Knopf, "3d object reconstruction using geometric computing," in *Geometric Modeling and Imaging--New Trends*, 2006, pp. 9-14.

[94] T. P. Thomas, D. D. Anderson, A. R. Willis, P. Liu, M. C. Frank, J. L. Marsh, *et al.*, "A computational/experimental platform for investigating three-dimensional puzzle solving of comminuted articular fractures," *Computer Methods in Biomechanics and Biomedical Engineering,* vol. 14, pp. 263-270, 2011.

[95] G. K. Knopf and J. Kofman, "Surface reconstruction using neural network mapping of range-sensor images to object space," *Journal of Electronic Imaging,* vol. 11, pp. 187-194, 2002.

[96] B. J. Brown, C. Toler-Franklin, D. Nehab, M. Burns, D. Dobkin, A. Vlachopoulos, *et al.*, "A system for high-volume acquisition and matching of fresco fragments: Reassembling theran wall paintings," *ACM Transactions on Graphics,* vol. 27, pp. 1-9, August 2008.

[97] M. S. Sağıroğlu and A. Erçil, "A texture based approach to reconstruction of archaeological finds," in *Symposium on Virtual Reality, Archaeology, and Cultural Heritage*, 2005.

[98] M. S. Sagiroglu and A. Ercil, "A texture based matching approach for automated assembly of puzzles," in *International Conference on Pattern Recognition*, Hong Kong, 2006, pp. 1036 - 1041.

[99] G. Papaioannou and E. A. Karabassi, "On the automatic assemblage of arbitrary broken solid artefacts," *Image and Vision Computing,* vol. 21, pp. 401-412, May 2003.

[100] G. Oxholm and K. Nishino, "Aligning surfaces without aligning surfaces," in *Proceedings of the IEEE Workshop on Applications of Computer Vision*, 2011, pp. 174-181.

[101] K. Son, E. B. Almeida, and D. B. Cooper, "Axially symmetric 3d pots configuration system using axis of symmetry and break curve," in *IEEE Conference on Computer Vision and Pattern Recognition*, Portland, OR, 2013, pp. 257 - 264.

[102] T. L. Faber and E. M. Stokely, "Orientation of 3-d structures in medical images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 10, pp. 626-633, 1988.

[103] C. Lo and H. Don, "3d moment forms: Their construction and application to object identification and positioning," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 11, pp. 1053-1064, 1989.

[104] M. Trummer, H. Suesse, and J. Denzler, "Coarse registration of 3d surface triangulations based on moment invariants with applications to object alignment and identification," in *IEEE International Conference on Computer Vision*, 2009, pp. 1273-1279.

[105] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software,* vol. 22, pp. 469-483, 1996.

[106] Z. Yang and F. Cohen, "Image registration and object recognition using affine invariants and convex hulls," *IEEE Transactions on Image Processing*, vol. 8, 1999.

[107] Z. Yang and F. Cohen, "Cross-weighted moments and affine invariants for image registration and matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, 1999.

[108] F. Cohen, Z. Liu, and T. Ezgi, "Virtual reconstruction of archeological vessels using expert priors and intrinsic differential geometry information," *Computers & Graphics*, vol. 37, pp. 41-53, 2013.

[109] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*: John Wiley & Sons, 2004.

[110] F. Cohen, Z. Zhang, and Z. Liu, "Mending broken vessels a fusion between color markings and anchor points on surface breaks," *Multimedia Tools and Applications*, pp. 1-24, 2014.

VITA

Zhongchuan Zhang received the B.S. degree in Optoelectronic Science and Technology from Nankai University, Tianjin, China, in 2007, and M.S. degree in Optical Engineering from Tianjin University, Tianjin, China, in 2009. He joined the Department of Electrical and Computer Engineering at Drexel University in September 2009, and studied with Dr. Fernand Cohen. During his Ph.D. study, he focused on 3D object tracking and 3D surface alignment. His research interests include computer vision, image processing and pattern recognition. He served as a teaching and research assistant at Drexel University. As a teaching assistant, he participated in teaching and developing the graduate/undergraduate level image processing and DSP courses. As a research assistant, he has published several papers at prestigious journals and conferences including MTAP, CVPR, VISAPP, ICDSC, ICMCS and ICIAP.