Multi-view Approaches to Tracking, 3D Reconstruction and Object Class Detection

by

Saad M. Khan
B.Sc., Ghulam Ishaq Khan Institute, 2003
M.Sc., University of Central Florida, 2007

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the School of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2008

Major Professor: Mubarak Shah

© 2008 by Saad M. Khan

Abstract

Multi-camera systems are becoming ubiquitous and have found application in a variety of domains including surveillance, immersive visualization, sports entertainment and movie special effects amongst others. From a computer vision perspective, the challenging task is how to most efficiently fuse information from multiple views in the absence of detailed calibration information and a minimum of human intervention. This thesis presents a new approach to fuse foreground likelihood information from multiple views onto a reference view without explicit processing in 3D space, thereby circumventing the need for complete calibration. Our approach uses a homographic occupancy constraint (HOC), which states that if a foreground pixel has a piercing point that is occupied by foreground object, then the pixel warps to foreground regions in every view under homographies induced by the reference plane, in effect using cameras as occupancy detectors. Using the HOC we are able to resolve occlusions and robustly determine ground plane localizations of the people in the scene. To find tracks we obtain ground localizations over a window of frames and stack them creating a space time volume. Regions belonging to the same person form contiguous spatio-temporal tracks that are clustered using a graph cuts segmentation approach.

Second, we demonstrate that the HOC is equivalent to performing visual hull intersection in the image-plane, resulting in a cross-sectional slice of the object. The process is extended to

multiple planes parallel to the reference plane in the framework of plane to plane homologies. Slices from multiple planes are accumulated and the 3D structure of the object is segmented out. Unlike other visual hull based approaches that use 3D constructs like visual cones, voxels or polygonal meshes requiring calibrated views, ours is purely-image based and uses only 2D constructs i.e. planar homographies between views. This feature also renders it conducive to graphics hardware acceleration. The current GPU implementation of our approach is capable of fusing 60 views (480x720 pixels) at the rate of 50 slices/second. We then present an extension of this approach to reconstructing non-rigid articulated objects from monocular video sequences. The basic premise is that due to motion of the object, scene occupancies are blurred out with non-occupancies in a manner analogous to motion blurred imagery. Using our HOC and a novel construct: the temporal occupancy point (TOP), we are able to fuse multiple views of non-rigid objects obtained from a monocular video sequence. The result is a set of blurred scene occupancy images in the corresponding views, where the values at each pixel correspond to the fraction of total time duration that the pixel observed an occupied scene location. We then use a motion de-blurring approach to de-blur the occupancy images and obtain the 3D structure of the non-rigid object.

In the final part of this thesis, we present an object class detection method employing 3D models of rigid objects constructed using the above 3D reconstruction approach. Instead of using a complicated mechanism for relating multiple 2D training views, our approach establishes spatial connections between these views by mapping them directly to the surface of a 3D model. To generalize the model for object class detection, features from supplemental

views (obtained from Google Image search) are also considered. Given a 2D test image, correspondences between the 3D feature model and the testing view are identified by matching the detected features. Based on the 3D locations of the corresponding features, several hypotheses of viewing planes can be made. The one with the highest confidence is then used to detect the object using feature location matching. Performance of the proposed method has been evaluated by using the PASCAL VOC challenge dataset and promising results are demonstrated.

Dedicated to my mother Yasmin Masood Khan

and in memory of my father Anwar Masood Khan.

Acknowledgments

I would like to thank my research advisor *Dr. Mubarak Shah* for his support and encouragement without which this research would not have been possible. I would also like to thank my colleagues and research collaborators *Dr. Pingkun Yan, Fahd Rafi, Kevin Boulanger, Pavel Babenko* and rest of the members of the UCF computer vision lab for their assistance in countless experiments and fruitful discussions. Finally my gratitude to the dissertation committee members *Dr. Steve Ebert, Dr. Marshall Tappen and Dr. Annie Wu.*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Background and Motivation

The ultimate goal of automated video surveillance is to detect, track and perhaps reconstruct objects of interest in the scene, enabling higher-level analysis like monitoring patterns of normal and abnormal behavior. As video surveillance becomes more ubiquitous, it has found application in a variety of scenarios, e.g., airport security, subway/railway stations, sports events, shopping malls, parking lots and along installations of national security. Many of these sites are heavily crowded adding an extra challenge for accurate and precise surveillance.

Tracking multiple people accurately in cluttered and crowded scenes is a challenging task primarily due to occlusion between people. If a person is visually isolated (i.e. neither occluded nor occluding another person in the scene) it is much simpler to perform the tasks of detection and tracking. This is because the physical attributes of the person's foreground blob like color distribution, shape and orientation remain largely unchanged as he/she moves. With increasing density of objects in the scene inter object occlusions increase. A foreground blob is no longer guaranteed to belong to a single person and may in fact belong to several people in the scene. Even worse, a person might be completely occluded by other people. Under such conditions of limited visibility and clutter it might be impossible to detect and

track multiple people using only a single view. The logical step is to try and use multiple views of the same scene in an effort to recover information that might be missing in a particular view.

This approach can also significantly help in determining the type or category of a detected object. An object may look drastically different from front-on when compared with a profile view. As a result a classifier trained to learn the appearance of an object from one view may perform poorly in detecting the same object from a different viewpoint. The challenge is, therefore, to design an algorithm capable of learning the variance of object appearance across different views and successfully detect the object in any view.

A parallel problem in visual scene analysis, which is gaining popularity due to availability of multiple-views in surveillance setups, is the recovery of three-dimensional structure and shape of scene objects. The problem of recovering three-dimensional shape from images is an interesting challenge because it is an every day task encountered by humans and is solved subconsciously. There is extensive literature on the recovery of structure from images which has found applications in a variety of domains including but not exclusive to reverse engineering three-dimensional shapes for manufacturing, visualization from arbitrary viewpoint e.g. virtual tourism, virtual reality and video matting, etc. A basic requirement for multi-view 3D reconstruction approaches is the availability of camera calibration parameters. This requires carefully setting up cameras in known and rigid configurations (stationary cameras) thus limiting the applicability to laboratory conditions or expensive, elaborate

setups. For the use of this technology in the surveillance domain the algorithms need to be able to handle non-stationary cameras from arbitrary viewpoints.

1.2 Goals

The goal of our work is to *track*, *reconstruct* and *recognize* objects in complex situations using multi-view data. More specifically:

- To localize and track multiple possibly inter-occluded objects in the scene.
- To reconstruct the 3D structure of stationary scene objects using multiple un-calibrated views.
- To reconstruct non-stationary, articulated objects in monocular video.
- Object class/category detection from arbitrary view points.

1.3 Outline of this Research

In this thesis, we propose multi-view approaches to tracking, reconstructing and recognizing scene objects in crowded and cluttered scenes. Unlike other multi-view approaches we develop purely image-based solutions thus eliminating the need for the difficult task of calibrating cameras. This is achieved by exploiting 2D planar constructs like planar homographies induced by planar structures in the scene. In our analysis we assume that atleast one reference scene plane is visible in the views. This is a reasonable assumption in typical surveillance installations monitoring people in busy crowded places where usually the

ground plane or a planar structure like a building wall is visible. Such planar structures usually occupy a large enough image region to be automatically detected and aligned using robust methods of locking onto the dominant planar motion.

1.3.1 Homographic Occupancy Constraint (HOC) for Localization and Tracking

For tracking we are interested in situations where the scene is sufficiently dense that partial or total occlusions are common and it can not be guaranteed that any of the objects will be visually isolated. Figure 1.1 shows some examples of a crowded scenes that we used to test our approach. Notice that very few people are viewed in isolation and there are cases of near total occlusion.

In our approach we do not use color models or shape cues of individual people. We neither detect nor track objects in any single camera, or camera pair; rather evidence is gathered from all the cameras into a synergistic framework and detection and tracking results are propagated back to each view. Our method of detection and occlusion resolution is based on geometrical constructs and only requires the distinction of foreground from background, which are obtained using standard background modelling techniques. At the core of our method is a novel planar homographic occupancy constraint (HOC) [1] that combines foreground likelihood information (probability of a pixel in the image belonging to the foreground) from different views to resolve occlusions and determine regions on scene planes that are occupied by people. The HOC interprets foreground as scene occupancy by non-background objects (in-effect using cameras as occupancy sensors) and states that

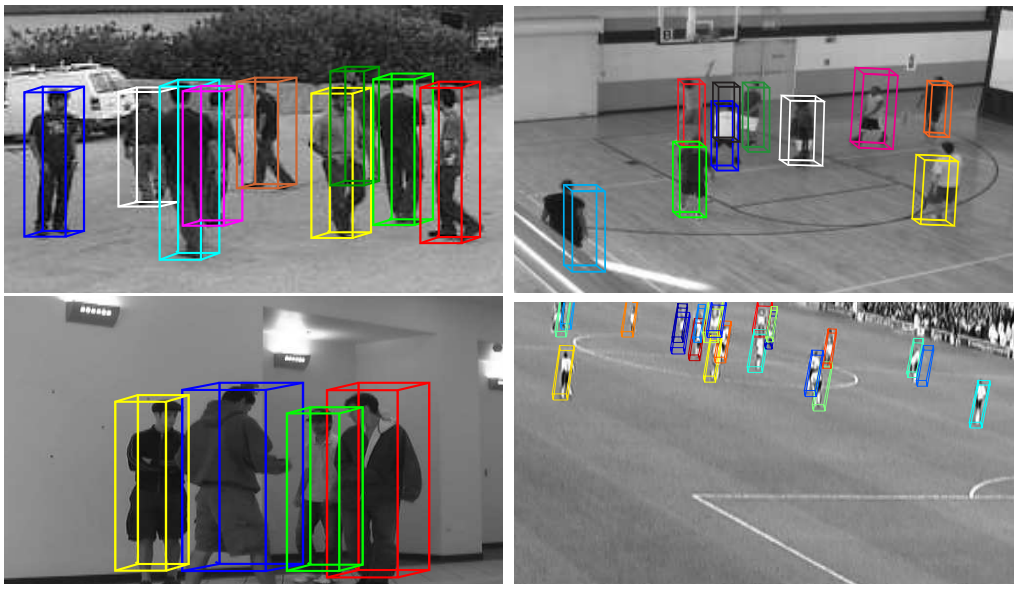


Figure 1.1: Examples of cluttered and crowded scenes used to test our approach. For illustration only one view for each scene is shown. Notice the occlusions. The density is such that most people are not visually isolated, they are either occluded or occluding other people in the scenes. There are also cases of near total occlusion yet using our approach we are able to detect and track individual people.

pixels corresponding to occupancies on a reference plane will consistently warp (under homographies of the reference plane), to foreground regions in every view. The reason we use foreground likelihood maps instead of binary foreground maps is to delay the thresholding step to the last possible stage.

To track, we obtain object scene occupancies for a window of time, and stack them together creating a space-time volume. Occupancies belonging to the same person form contiguous spatio-temporal regions that are clustered using a graph cuts segmentation ap-

proach. This is achieved by designing an energy functional that combines scene occupancy information and spatio-temporal proximity. The energy functional is minimized over the spatio-temporal grid using graph cuts that results in the segmentation of contiguous spatio-temporal clusters. Each cluster is the track of a person and a slice in time of this cluster gives the tracked location.

1.3.2 Homographic Occupancy Constraint on Multiple Scene Planes and 3D Reconstruction

The HOC used to localize and track scene objects is shown to be equivalent to voxel-based visual hull intersection on a *plane*, delivering a 2D grid of space-occupancies. This is interpreted as a cross-sectional slice of the scene objects cut out by the reference plane used for the HOC. By considering every 3D object to be composed of an infinite number of cross-sectional slices, with the frequency at which we sample the slices being a variable determining the granularity of the reconstruction. The problem of determining a particular slice is solved using the HOC.

The HOC is not only limited to a reference scene plane, but is extended to multiple planes parallel to the reference plane to robustly reconstruct and localize scene objects. Each plane gives a cross-sectional slice of the object that is pieced together slice by slice in a manner not too different from a CAT (Computed Axial Tomography) procedure in medical imagery. This extension to multiple planes also significantly reduces the false positives and missed

Figure 1.2: An example of the reconstruction results of our approach. Different view points of the 3D reconstruction are shown.

detections due to artifacts like shadows, or when it cannot be guaranteed that a single plane will be consistently occupied by scene objects.

1.3.3 Reconstruction of Non-stationary Articulated Objects in Monocular Sequences

The HOC as developed and utilized in the conventional sense can be used for reconstruction and localization in a monocular video sequence if the object is stationary (rigid). In some practical scenarios this is not the case, we therefore, extend our approach to reconstruct non-stationary, articulated objects in monocular video. We introduce the concept of *motion blurred scene occupancies*, a direct analogy of motion blurred images but in a 3D object scene occupancy space resulting from the motion/deformation of the object. Our approach starts with an image based fusion step that combines color and silhouette information from

multiple views. To this end we propose to use a novel construct: the temporal occupancy point (TOP), which is the estimated 3D scene location of a silhouette pixel and contains information about duration of time it is occupied. Instead of explicitly computing the TOP in 3D space we directly obtain it's imaged(projected) locations in each view. This enables us to handle monocular video and arbitrary camera motion in scenarios where complete camera calibration information may not be available. The result is a set of blurred scene occupancy images in the corresponding views, where the values at each pixel correspond to the fraction of total time duration that the pixel observed an occupied scene location. We then use a motion de-blurring approach to de-blur the occupancy images. The de-blurred occupancy images correspond to a silhouettes of the mean/motion compensated object shape and are used to obtain a visual hull reconstruction of the object. We show promising results on monocular datasets of deforming objects where traditional visual hull intersection approaches fail to reconstruct the object correctly.

1.3.4 Object Class Detection from Arbitrary View

Finally, we present an object class detection method based on 3D object modeling. Instead of using a complicated mechanism for relating multiple 2D training views, our approach establishes spatial connections between these views by mapping them directly to the surface of 3D model. The 3D shape of an object is reconstructed using our previously described homographic framework, from a set of model views around the object. Features are computed in each 2D model view and mapped to the 3D shape model. To generalize the model for

object class detection, features from supplemental views are also considered. A codebook is constructed from all of these features and then a 3D feature model is built. Given a 2D test image, correspondences between the 3D feature model and the testing view are identified by matching the detected features. Based on the 3D locations of the corresponding features, several hypotheses of viewing planes can be made. The one with the highest confidence is then used to detect the object using feature location matching. Performance of the proposed method has been evaluated by using the PASCAL VOC challenge dataset and promising results are demonstrated.

1.4 Organization of Thesis

The rest of this thesis is structured as follows. In Chapter 2 we discuss related work. Chapter 3 details the homographic occupancy constraint used to perform tracking. In Chapter 4 we develop the relationship between the homographic occupancy constraint and plane based visual hull intersection. We extend our approach to multiple planes and use the formulation for 3D reconstruction and more robust tracking. In Chapter 5 we present our approach to extend our 3D reconstruction algorithm to non-stationary, articulated objects. Chapter 6 describes our arbitrary view object class detection method that builds on the multi-view 3D reconstruction work we have done. We conclude this thesis in chapter 7 and provide directions for future work in chapter 8.

CHAPTER 2
RELATED WORK

In this chapter we provide context for this research in the backdrop of previous work. We evaluate related work in the areas of tracking, 3D reconstruction and object detection and discuss their advantages and potential deficiencies related to problems that we aim to solve. We will present some of them in detail or refer to others in the rest of this thesis when necessary.

2.1 Multiple Object Tracking

Broadly speaking the literature on tracking multiple occluding targets in cluttered scenes can be divided into two categories: single view approaches and Multi-view approaches, some of their representative techniques are described below. For a detailed review of the state of the art in tracking the reader is referred to the recent survey by Yilmaz et al. [20].

2.1.1 Single View Approaches

There is extensive literature on detection and tracking of multiple targets in a single-camera. This approach has the inherent advantage of simple and easy deployment, but has limited ability to handle occlusions involving several objects due to the limited field of view of a single camera.

Blob-tracking is a popular low cost approach for tracking objects [16] [5]. It entails extracting blobs in each frame and tracking is performed by associating blobs from one frame to the next. The *BraMBLe* system [8], for example, is a multi-blob tracker that generates a blob-likelihood based on a known background model and appearance models of the tracked people. Its performance degrades when multiple objects merge into one blob due to proximity or occlusions. Alternate approaches maintain explicit object states with position, appearance, and shape. Zhao and Nevatia [6] present interesting results of tracking multiple people in a single camera. They use articulated ellipsoids to model human shape, color histograms to model different peoples appearance, and an augmented Gaussian distribution to model the background for segmentation. Once pixels corresponding to the moving head in the scene are detected, a principled MCMC approach is used to maximize the posterior probability of a multi-person configuration.

Okuma et al. [9] propose an interesting combination of Adaboost for object detection and particle filters for multiple-object tracking. The combination of the two approaches leads to fewer failures than either one on its own, as well as addressing both detection and consistent track formation in the same framework. Brostow et al. [10] present a probabilistic framework for the clustering of feature point trajectories to detect individual pedestrians in crowds. They hypothesize that pairs of points that appear to move together are likely to be part of the same individual and as such be used for the detection and eventual tracking. This assumption appears to hold for somewhat overhead views but the plane-parallax induced in oblique views results in a degradation of performance. These and other similar approaches

like [11, 12, 13, 14, 15] skip the modeling of articulations in favor of appearance models trained for specific un-occluded views of their respective subjects. As a result they are challenged by fully and partially occluding objects, as well as appearance changes.

For handling occlusions, a number of monocular tracking techniques have been devised. The typical approach is to detect the occurrence of occlusion by blob merger [16]. The methods for tracking feature points simply detect the occlusion of a feature point as the disappearance of the point being tracked [17]. In recent years, tracking techniques using object contours [19] [18] and appearances [22] [21], which represent and estimate occlusion relationships between objects by using hidden variables of depth ordering of objects toward the camera, have been proposed. Wu et al. [22] incorporate an extra hidden process for occlusion into a dynamic Bayesian network, and rely on the statistical inference of the hidden process to reveal occlusion relations. Senior et al. [24] use appearance models to localize objects and use disputed pixels to resolve their depth ordering during occlusions. However the system cannot maintain object identity after occlusions. Jojic et al. [25] and Tao et al. [26] both model videos as a layered composition of objects and use EM to infer objects appearances and motions. Recently, Perera et al. [27] proposed a two-stage framework (one-to-one correspondences followed by a split and merge analysis) for linking tracks across occlusions. Most of the above mentioned approaches rely on partial observations, which makes it difficult to handle full occlusions. In addition, small and consistent motions are assumed to enable the prediction of motion patterns through occluded views. This causes problems in dealing with long periods of occlusions under unpredictable motions. In spite

of the work done to date, we believe monocular methods have limited ability to handle occlusions involving several objects, generally two or three, because the single viewpoint is intrinsically unable to observe the hidden object parts.

2.1.2 Multi-Camera Approaches

The use of multiple cameras soon becomes necessary when one wishes to accurately detect and track multiple occluding people and compute their precise locations in a complex environment. Multi-view tracking techniques intend to decrease the hidden regions and provide 3D information about the objects and the scene by making use of redundant information from different view-points.

In [28], Kelly et al. constructed a 3D environment model using calibrated cameras. Humans were modelled as a collection of these voxels to resolve the camera-handoff problem. Jain and Wakimoto, [31], also assumed calibrated cameras to obtain 3D locations of each object in an environment model for Multiple Perspective Interactive Video. Although the problem of tracking objects across cameras was not explicitly addressed, several innovative ideas were proposed, such as choosing the best view given a number of cameras and the concept of interactive television. These works were characterized by the use of environment models, and calibrated cameras. Multi-target tracking by association across multiple views was addressed in its own right, in a series of papers from the latter half of the 90s. In [30], Nakazawa et al. constructed a state transition map that linked regions observed by one or more cameras, along with a number of action rules to consolidate information between

cameras. Orwell et al. [32] present a tracking algorithm to track multiple objects in multiple views using 'color' tracking. They model the connected blobs obtained from the background subtraction using color histogram techniques and employ them to match and track objects. Cai and Aggarwal [33] extend a single-camera tracking system by starting with tracking in a single camera view and switching to another camera when the system predicts that the current camera will no longer have a good view of the subject. Spatial matching was based on the Euclidean distance of a point with its corresponding epipolar line. In [34], individuals are tracked both in image planes and top view using a combination of appearance and motion models. Bayesian Networks were used in several papers as well. In [36], Chang and Gong used Bayesian networks to combine geometry (epipolar geometry, homographies and landmarks) and recognition (height and appearance) based modalities to match objects across multiple sequences. Bayesian networks were also used by Dockstader and Tekalp in [35], to track objects and resolve occlusions across multiple calibrated cameras. Integration of stereo pairs is another popular approach, adopted by [38] [39] [37] amongst others. Krumm et al. [38] use stereo cameras and combine information from multiple stereo cameras in 3D space. They perform background subtraction and then detect human-shaped blobs in 3D space. Color histograms are created for each person and are used to identify and track people. Mittal et al. [39] use a similar method to combine information in pairs of stereo cameras. Regions in different views are compared with each other and back-projection in 3D space is done in a manner that yields 3D points guaranteed to lie inside the objects.

Even though these methods attempt to resolve occlusions, the underlying problem of using features (appearance templates, blob shapes) that might be corrupted due to occlusions remains. Secondly, occlusion reasoning in these approaches is typically based on temporal consistency in terms of a motion model, whether it is Kalman filtering or more general Markov models. As a result, these approaches may not always be able to recover if the process starts diverging. The scenes shown in figure 1.1 would be difficult to resolve for most of these methods. Not only are there cases of near total occlusion, the people are dressed in very similar colors. Using blob shapes or color distributions for region matching across cameras may lead to incorrect segmentations and detections.

The homographic occupancy constraint (HOC) [1] presented in this thesis fuses information from multiple views using sound geometrical constructs and resolves occlusions by localizing people on multiple scene planes. In our approach the only time appearance information is used is to detect foreground from background, making it robust to appearance occlusions in crowded scenes. In essence using foreground information in multiple views, we attempt to find image locations of scene points that are guaranteed to be occupied by people. These occupancies are then used to resolve occlusions and track multiple people. In this spirit the work by Mittal et al. [39], Leibe et al. [40], Franco et al. [41] and the parallel work on range sensor based occupancy grids for robot navigation is quite relevant [42] [43]. But unlike these approach that fuse information in 3D space requiring calibrated cameras, our approach is completely image-based and requires only 2D constructs to perform fusion in the image plane without requiring to go in 3D space. Some approaches have been proposed that

do not require prior calibration of cameras, but instead minimal relative camera information is learned [45]. In [44], Khan et al. proposed an approach that avoided explicit calibration of cameras and instead used constraints on the field of view lines between cameras, learnt during a training phase, to track objects across the cameras. These and similar techniques track objects in individual un-calibrated views and then create associations across views for better situation awareness purposes. However, since they rely on tracking in individual cameras, their approach suffers due to increased density of objects in crowded scenes. We neither localize nor track people from any single camera, or camera pair; rather evidence is gathered from all the cameras into a unified, synergistic framework where detection and tracking are performed simultaneously. The novelty of our approach is to perform global trajectory optimization on scene occupancy probabilistic data from multiple time frames, thereby seamlessly combining the task of detection and tracking.

2.2 3D Reconstruction

Three-dimensional reconstruction is one of the oldest problems in computer vision. This section presents a short review of the field of multi-view 3-D reconstruction, and discusses in particular how the underlying 3-D representation plays a important role in the types of scenes that can be reconstructed. Visual hull based methods are of particular relevance to this thesis, and are reviewed in detail in Section 2.2.2.

2.2.1 Stereo Reconstruction

The earliest attempt to solve the stereo problem, by Marr and Poggio [60], was based upon human-stereopsis. As the field has progressed the emphasis has moved further away from understanding human vision towards the more general problem of recovering three-dimensional shape. There are two very good reviews of early vision; the first is by Barnard and Fischler [61]and covers the early 70s and 80s. The second is by Dhond and Aggarwal [62] and covers the late 80s. Stereo algorithms typically represent shape using a depth map, or disparity map, which is also known as a 2.5-D sketch (Marr and Poggio [60]]). In this representation depth $z(x, y)$, or disparity $d(x, y)$, is represented as a function of the pixel co-ordinates (x, y) of a reference image. The disparity representation $d = x_L - x_R$ is useful because it stores the change in position of an image feature between the left and right images without requiring triangulation to obtain depth. The 2.5-D representation proves to be a limiting factor in most stereo algorithms as it cannot represent arbitrary shapes. In next section visual hull representations are considered as they can represent more general shapes.

Stereo algorithms operate using either edge features, or dense correlation, or both. Edge matching is useful because the intensity edges, in an image, describe the important geometry of the image and the areas between the intensity edges are often uniform. One technique for locating these edges is the Marr Hildreth edge operator [63], which has been extensively used to obtain edges from stereo images (e.g. Grimson [68], Mayhey and Frisby [64] and Ayache and Faverjon [69]). The Marr-Hildreth [63] operator convolves the image with a mask approximating the Laplacian of a Gaussian and then labels all the zero crossings as

edges. Other popular edge detectors are Canny [65] and Deriche [66]. One of the difficulties with edge-based stereo is that correspondences are only known along the edges. Assumptions such as continuity (Marr and Poggio [60]), can be used to fill in the gaps between the edges, but these assumptions are only true for specific image sequences.

When finding correspondences between two images, there are a number of constraints that can be applied to simplify and regularize the correspondence problem. The epipolar constraint, Faugeras [70] and Hartley [71], on the other hand is a geometric constraint and is always valid. It is useful because it reduces the dimensionality of the search space, when matching image features. A point in an image defines a ray in space, and the corresponding three-dimensional point must lie along this ray. From any other viewpoint this ray will appear as a line, so the corresponding feature in the second image must lie along this epipolar line. This epipolar constraint means that it is only necessary to perform a 1-D search, not a 2-D search, to find each corresponding point feature. Features are usually matched using Dynamic Programming (Baker and Binford [73]) or they can be matched using relaxation labelling (Rosenfeld, Hummel and Zucker [74]). Dynamic programming has been used to match edges, and to obtain dense stereo correspondences, but can only be applied on a scanline-by-scanline basis. More recently, the inter-scanline consistency problem has been solved by Boykov, Veksler and Zabih [76] and Roy and Cox [75], who have used graph cuts to find a global solution to the correspondence problem. This is an elegant solution, but still requires the model to be stored as a 2.5-D sketch. The implementation by Ishikawa and Geiger [77] is similar but it optimizes a surface in 3-D rather than a disparity map. This is an

improvement, but has more in common with the volumetric techniques which are presented next.

2.2.2 Visual Hull Based Methods

Matching edges and intensities is not the only method for obtaining three dimensional shape from image sequences. When a three-dimensional object is viewed in an image, the outline (or profile) provides very strong information about the shape of the object. From a single viewpoint the silhouette defines a cone in space and the 3-D object must lie within this cone. This must be true for any number of images, so the 3-D shape must lie within the intersection of all these cones. The term visual hull was defined by Laurentini [115] and is the shape obtained in the limit, by adding an infinite number of silhouettes from all locations outside the convex hull of the object, and must contain the convex hull.

Many different shape representations have been considered and have been reviewed by Szeliski [128] and Dyer [78]. The first group of algorithms that are considered are the polygon-based representations. These provide a Euclidean (or projective) representation for space. The polygons can be obtained automatically by matching image features (e.g. Beardsley, Torr and Zisserman [80]), or with user assistance (Debevec et. al. [81], and Cipolla and Robertson [86]), or by combining the results of many stereo pairs (e.g. Koch and Pollefeys [88], [87]). Since the polygon representations are very general in shape, it is always useful to include prior knowledge about the scene. This is especially useful in the case of architecture (e.g. Torr, Dick and Cipolla [92]), where windows and doors are often regular

and can be represented by parametric shape models. Polygon-based reconstructions can be made significantly more realistic by using photographic and bas-relief texture. This was demonstrated by Debevec, Taylor and Malik [81], and further developed by Baker, Szeliski and Anandan [82] who used layered depth images to efficiently represent 3-D scenes. Polygon models are efficient to store and render, when the scene contains large planar surfaces, but there are many different ways to partition any one scene into polygons. In addition, a reconstruction often requires a shape to be closed and enforcing this constraint with polygons can be difficult.

Another class of algorithms use the volumetric representations for which there is an excellent review by Dyer [78]. The volumetric approaches may use voxels (Szeliski and Golland [93], Seitz and Dyer [94], Slabaugh et. al. [97]), or level sets (Sethian [95]). Level sets were introduced into Computer Vision by Faugeras and Keriven [96]. The voxel-based algorithms make no attempt to model the continuity of shape, rather, they model the volume as an array of 3- D voxels. By not making assumptions about planarity and continuity, the voxel algorithms are able to cope with significantly more complex structures. The difficulty with these approaches, however, is that by ignoring the regularizing assumptions, they become more susceptible to noise.

Several methods have also been proposed to bypass silhouette estimation altogether, as many algorithms reconstruct the scene structure based only on photometric information [89] [90] [119]. Crucially, this class of methods must deal with the visibility of points on the object's surface (occlusion reasoning) making them more complicated and computationally

expensive. This is why there are still many situations where silhouette-based methods are preferred e.g. VR platforms or real-time interactive systems. For further details the reader is directed to an excellent recent survey of the area [125].

The common feature amongst all these methods is the requirement of fully calibrated views and the use of 3D constructs like voxels or visual cones being intersected in the 3D world. Herein lies the novelty of our approach. We present a completely image-based approach that uses only 2D constructs like planar homographies for silhouette fusion *in the image plane* without requiring to go in 3D space, thereby circumventing the difficult task of fully calibrating views.

2.3 Object Class Detection

As the approaches for recognizing an object class/type from some particular viewpoint or detecting a specific object from an arbitrary view are advancing toward maturity [143, 159, 161], solutions to the problem of object class detection using multiple views are still relatively far behind. Researchers in computer vision have studied the problem of arbitrary view object class detection resulting in the following major directions.

One popular approach is to use features that are by themselves invariant to pose transformations. The basic idea is to identify image-based measures that remain constant as a function of viewing direction, and use them as a signature that identifies an object. A variety of features invariant under affine transformations have been proposed in literature. For instance the four-point cross-ratio and other more sophisticated algebraic invariance [144]

have been used for object recognition [141][153]. Other researchers have attempted to use features that are nearly invariant for more general transformations [147]. One limitation of this approach is the difficulty in finding a sufficient number of invariant features for reliable recognition, particularly when objects that are similar in overall shape (such as cows from horses) have to be discriminated.

Another path attempts to use increasing number of local features by applying multiple feature detectors simultaneously [140, 148, 163, 164, 165]. It has been shown that the recognition performance can be benefited by providing more feature support. However, the spatial connections of the features in each view and/or between different views have not been pursued in these works. Most of the proposed methods apply several single-view detectors independently and then combine their responses via some arbitration logic. At best, features are shared among the different single-view detectors to limit the computational overload [151]. These connections can be crucial in object class detection tasks. In contrast, we do not rely on single-view detectors working independently, but develop a single integrated multi-view detector that we call the 3D feature model. Our approach integrates 3D shape models of objects with appearance features obtained from a large number of training images with varying view points and object class instances.

In the context of face detection, some more sophisticated schemes have been proposed. For instance, [152] studies the trajectories of faces in linear PCA feature space as they rotate, while a detector pyramid is used in [154]. Fan and Lu [157] propose a multi-view face recognition approach that combines feature selection with multi-class classification based

on SVM. This yields a discriminative set of features and consequently good recognition results without splitting the training data in separate views. However, these methods are specialized for faces and do not directly generalize to generic object detection. Several researchers have used PCA to achieve pose invariance for instance, Murase and Nayar [155] acquired images of several objects every four degrees. From these images they constructed an eigenspace representation for a given object, and used it for recognizing the object in different poses. Similarly, Pentland et al. [156] used only five views of faces (between frontal and profile) for arbitrary view recognition. Though the approach performs well when the view to be recognized is at the same orientation as the training set, the accuracy dropped quickly when interpolation or extrapolation between views was required. Recently, much attention has been drawn to this direction related to multiple views for object class detection [146, 149, 150].

Closely related to our approach is the work of Bart et al. [158]. Similar to our approach, they developed a system to recognize specific instances of an object class under arbitrary viewpoint given just a single example view. This is achieved by using 'extended fragments', learnt from short video sequences showing other instances of the same class. Yet, in [158] they are used to link only two viewpoints (frontal and 60 degrees), and the application domain (faces) is also different from ours. Whereas in our approach the 3D feature model links viewpoints from any pose. Most recently, Thomas et al. [166] developed a single integrated multi-view detector that accumulates evidence from different training views. Their work combines a multi-view specific object recognition system [159], and the Implicit Shape Model

for object class detection [161], where single-view codebooks are strongly connected by the exchange of information via sophisticated activation links between each other.

The main contribution of our work is an efficient object detection system capable of recognizing and localizing objects from the same class under different viewing conditions. We develop a unified method to relate multiple 2D views based on 3D object modeling. Having a full 3D model of an object alleviates the need to store multiple views, since novel views may be generated from such a model. Thus, 3D locations of the features are considered during detection and better accuracy is obtained.

2.4 Summary

In this chapter we have provided a detailed literature analysis of related work in fields of tracking, 3D reconstruction and object class detection. We have evaluated related work and discussed their advantages and potential deficiencies related to problems that we aim to solve. We have also provided brief descriptions of our solutions to the above mentioned problems and put them in context with previously published work. In the next four chapters we go into details of our approaches to tracking, 3D reconstruction and object class detection.

CHAPTER 3
TRACKING MULTIPLE PEOPLE USING A HOMOGRAPHIC OCCUPANCY CONSTRAINT

3.1 Introduction

Tracking multiple people accurately in cluttered and crowded scenes is a challenging task primarily due to occlusion between people. If a person is visually isolated (i.e. neither occluded nor occluding another person in the scene) it is much simpler to perform the tasks of detection and tracking. This is because the physical attributes of the person's foreground blob like color distribution, shape and orientation remain largely unchanged as he/she moves. Increasing the density of objects in the scene increases inter-object occlusions. A foreground blob is no longer guaranteed to belong to a single person and may belong to several people in the scene. Even worse, a person might be completely occluded by other people. Under such conditions of limited visibility and clutter it might be impossible to detect and track multiple people using only a single view. The next logical step is to use multiple views of the same scene in an effort to recover information that might be missing in a particular view.

In this chapter, we present a multi-view approach to detecting and tracking multiple people in crowded and cluttered scenes. We are interested in situations where the scene is sufficiently dense that partial or total occlusions are common and it can not be guaranteed that any person will be visually isolated. Figure 1.1 shows several examples of crowded scenes

that we used to test our approach. Notice that very few people are viewed in isolation and there are cases of near total occlusion. In our approach we neither detect nor track objects in any single camera, or camera pair; rather evidence is gathered from all the cameras into a synergistic framework and detection and tracking results are propagated back to each view. This is achieved using a homographic occupancy constraint [1] that is described next.

3.2 Homographic Occupancy Constraint

We begin with the basic notions of planar homographies. Let $p = (x, y, 1)$ denote the image location (in homogeneous coordinates) of a 3D scene point in one view and let $p' = (x', y', 1)$ be its coordinates in another view. Let H_π denote the homography induced by plane π between the two views. When the first image is warped toward the second image using the homography H, then the point p will move to p_w in the *warped image*:

$$p_w = (x_w, y_w, 1) \approx [H_\pi]p.$$

For scene points on the plane π, $p_w = p'$. For scene points off π, $p_w \neq p'$. The misalignment $p_w - p'$ is called the plane parallax. Geometrically speaking warping pixel p from the first image to the second using the homography H_π amounts to projecting a ray from the camera center through pixel p and extending it till it intersects the plane π at the point often referred to as the 'piercing point' of pixel p with respect to plane π. The ray is then projected from the piercing point onto the second camera. The point in the image plane of the second camera that the ray intersects is p_w. In effect p_w is where the image of the piercing point is

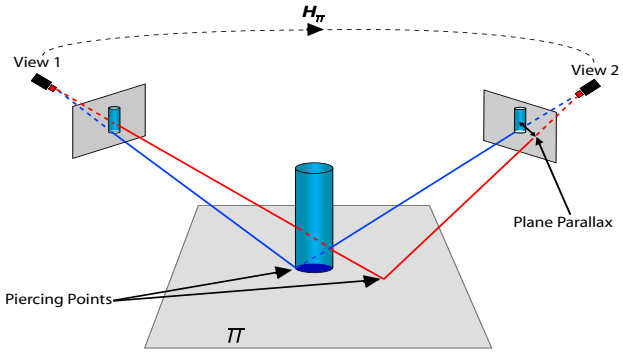


Figure 3.1: The figure shows a cylindrical object on a planar surface. The scene is being viewed by two cameras. H_π is the homography of the planar surface from view 1 to view 2. Warping a pixel from view 1 with H_π amounts to projecting a ray on to the plane at the piercing point and extending it to the second camera. Pixels that are image locations of scene points off the plane have plane parallax when warped. This can be observed for the red ray in the figure.

formed in the second camera. As can be seen in figure 3.1, scene points on the plane π have no plane-parallax while those off the plane have considerable plane-parallax.

Suppose a scene containing a reference plane is being viewed by a set of wide-baseline stationary cameras. The background models in each view are available and when an object appears in the scene it can be detected as foreground in each view using the background difference. Any scene point lying inside the foreground object in the scene will be projected to a foreground pixel in every view. The same is the case for scene points inside the object that lie on the reference plane, except however that the projected image locations in each view will be related by homographies induced by the reference plane. Now we can state the

following proposition:

Proposition 1 If $\exists P \in \mathbf{R}^3$ such that it lies on scene plane π and is inside the volume of a foreground object then, the image projections of the scene point P given by $p_1, p_2, \ldots, p_n$ in any n views satisfy both of the following:

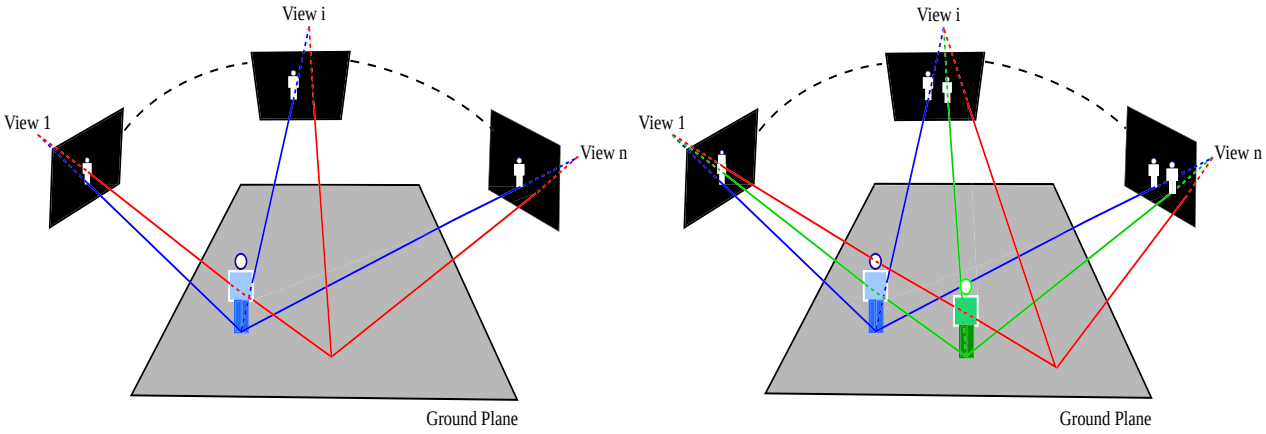
- $\forall_i$, if Ψ_i is the foreground region in view i then, $p_i \in \Psi_i$,
- $\forall_{i,j} p_i = [H_{i_\pi j}]p_j$, where $H_{i_\pi j}$ is the homography induced by plane π from view j to view i.

As discussed earlier warping a pixel from one image to another using a homography of a plane amounts to projecting a ray through the pixel onto the piercing point and then projecting it to the second camera center. If the ray projected through a pixel in a view intersects the reference plane inside a foreground object in the scene, it follows from **Proposition 1** that the pixel will warp to foreground regions in all views. This can be formally stated as follows:

Proposition 2 Let Φ be the set of all pixels in a reference view r and let $H_{i_\pi r}$ be the homography of plane π in the scene from the reference view to view i. If $\exists p \in \Phi$ such that the piercing point of p with respect to π lies inside the volume of a foreground object in the scene then $\forall_i p'_i \in \Psi_i$, where $p'_i = [H_{i_\pi r}]p$ and Ψ_i is the foreground region in view i.

We call **Proposition 2** the *homographic occupancy constraint (HOC)* [1]. Notice that the HOC does not distinguish between foregrounds in different views that may correspond to different objects. It is essentially using camera sensors as scene occupancy detectors with foreground interpreted as occupancy in the line of sight of the image sensor. And though the foreground regions associated across views may correspond to different scene objects (specifically the nearest foreground object along the line of sight of the particular image sensor), the HOC insures that they all correspond to the same scene occupancy.

This has the dual action of localizing people in the scene as well as resolving occlusion. To see this consider figure 5.1. Figure 5.1a shows a scene containing a person viewed by a set of cameras. The foreground regions in each view are shown as white on black background. A pixel that is the image of the feet of the person will have a piercing point on the ground plane (the reference plane for this example) that is inside the volume of the person. According to the HOC such a pixel will be warped to foreground regions in all views. This can be seen for the pixel in view 1 of figure 5.1a that has a blue ray projected through it. Foreground pixels that do not satisfy the HOC are images of points off the ground plane. Due to plane parallax they are warped to background regions in other views. This can be seen for the pixel with a red ray projected through it. Figure 5.1b shows how the HOC would resolve occlusions. The blue person is occluding the green person in view 1. This is apparent by the merging of their foreground blobs. In such a case there will be two sets of pixels in view 1 that satisfy the homoraphy constraint. The first set will contain pixels that are image locations of blue person's feet (same as in figure 5.1a). The other set of pixels are those that correspond to the



(a) (b)

Figure 3.2: The figure shows people viewed by a set of cameras. The views show the foreground detected in each view. For figure (a) the blue ray shows how the pixels that satisfy the HOC warp correctly to foreground in each view, while others have plane parallax and warp to background. Figure (b) demonstrates how occlusion is resolved in view 1. Foreground pixels that belong to the blue person but are occluding the feet region of the green person satisfy the HOC (the green ray). This creates seemingly a see through effect in view 1, where the feet of the occluded person can be detected.

blue person's torso region but are occluding the feet of the green person. Even though these pixels are image locations of points off the ground plane, they have piercing points inside a foreground object which in this case happens to be the green person. This process creates a seemingly see thorough effect detecting feet regions even if they are completely occluded by other people. Clearly having more people between the blue and the green person will not affect the localization of the green person on the ground plane.

It should be noted that the HOC is not limited to the ground plane and depending on the application any plane in the scene could be used. In the context of localizing people in a surveillance scenario the ground plane is typically a good choice if it is clearly visible. In other scenarios a building wall or any planar landmark can be used as the reference plane. In the next section we develop an operator that uses the HOC to localize people on a reference plane.

3.3 People Localization

Let $\Phi_1, \Phi_2, \ldots, \Phi_n$ be the images of the scene obtained from n uncalibrated cameras. Let Φ_r be a reference view. $H_{i\pi r}$ is homography of the reference plane π between the reference view and any other view i. Using homography $H_{i\pi r}$, a pixel p in the reference image is warped to pixel p'_i in image Φ_i. Let $x_1, x_2 \ldots, x_n$ be the observations in images $\Phi_1, \Phi_2, \ldots \Phi_n$ at locations $p'_1, p'_2 \ldots, p'_n$ respectively i.e $x_i = \Phi_i(p'_i)$. Let E be the event that pixel p has a piercing point inside a foreground object (i.e. p represents the reference plane π location of a foreground object in the scene). Given $x_1, x_2 \ldots, x_n$, we are interested in finding the

probability of event E happening, i.e $P(E \mid x_1, x_2 \ldots, x_n)$.

Using Bayes law:

$$P(E \mid x_1, x_2 \ldots, x_n) \propto P(x_1, x_2 \ldots, x_n \mid E)P(E). \tag{3.1}$$

The first term on the right hand side of equation 1 is the likelihood of making observation $x_1, x_2 \ldots, x_n$ given event E happens. By conditional independence we can write this term as:

$$P(x_1, x_2 \ldots, x_n \mid E) = P(x_1 \mid E) \times P(x_2 \mid E) \times \ldots \times P(x_n \mid E). \tag{3.2}$$

Now the HOC states that if a pixel has a piercing point inside a foreground object then it will warp to foreground regions in every view. Therefore it follows that:

$$P(x_i \mid E) \propto L(x_i), \tag{3.3}$$

where $L(x_i)$ is the likelihood of observation x_i belonging to the foreground. Plugging (3) into (2) and back into (1) we get:

$$P(E \mid x_1, x_2 \ldots, x_n) \propto \prod_{i=1}^{n} L(x_i). \tag{3.4}$$

The value of $P(E \mid x_1, x_2 \ldots, x_n)$ given by equation 4, represents the likelihood of the scene location being occupied by the foreground object. In effect, we are hypothesizing in the reverse direction i.e. reasoning about scene occupancies from the fusion of scene observations.

3.3.1 Modelling Clutter and Field of View Constraints

So far we have assumed the scene point under examination is inside the field of view of each camera, limiting our analysis to overlapping region of the multi-view setup. Also the fusion

operation in equation 3.4 assigns uniform prior precedence to each view. Due to the varying amounts of clutter in a particular view, the degree of confidence in foreground detection will be effected. Clutter may cause false detections or miss the foreground in some cases. Therefore, in this section we propose to use a measure of clutter, in order to weigh the foreground likelihood information detected from different views in our fusion model.

Schmieder and Weathersby [49] proposed the concept of a rms clutter metric of the spatial-intensity properties of the scene. Due to its robustness and applicability, it is one of the most commonly used clutter measures . Experimental results that have been reported in the literature [49] [50] that show a high correlation between the average target detection time by human subjects and the Schmieder and Weathersby (SW) clutter metric. The SW clutter metric is computed by averaging the variance of contiguous square cells over the whole image:

$$C = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\sigma_k^2}, \tag{3.5}$$

where σ_k^2 is the variance of pixel values within the kth cell and N is the number of cells or blocks the picture has been divided into. Typically, N is defined to be twice the length of the largest target (in our case humans) dimension. We compute the clutter metric for each view at each time instant on the *foreground likelihood maps* obtained from background modelling (for each pixel the likelihood of being foreground), therefore σ_k^2 in equation 3.5 is the variance of foreground likelihood values in the kth cell. Figure 3.3 shows some of the views of datasets used in our experiments (first row), their corresponding foreground likelihood maps (second row) and the SW clutter values obtained from them. As illustrated

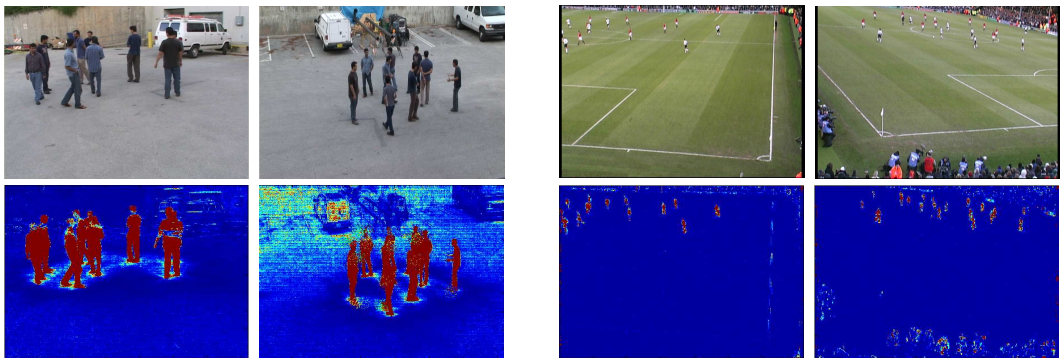


Figure 3.3: The first row shows images from two of our test sequences (2 views each). The second row shows foreground likelihood maps for views in the first row, where redder corresponds to greater foreground likelihood. The SW clutter metric is computed on these foreground likelihood maps. It can be visually corroborated that views with noisy foreground likelihood maps have higher clutter value.

by figure 3.3, the views with more noise and clutter in the foreground likelihood maps have a greater SW clutter metric value.

In order to assign higher confidence to foreground detected from views with lesser clutter we use the following method. For each foreground likelihood map i, we use clutter C_i, computed using equation 3.5 as its prior weight in the log-likelihood of the fusion operation in equation 3.4:

$$\log(P(X \mid x_1, x_2 \ldots, x_n)) \propto \sum_{i=1}^{n} \frac{1}{\tau C_i} \log(L(x_i)), \tag{3.6}$$

where $\tau = \sum_i \frac{1}{C_i}$, is a normalizing factor. The effect of modelling clutter on the performance of our approach is further discussed in the results and experiments section.

Though equation 3.6 ensures that the evidence from all available views is combined to maximize the certainty in the localization hypothesis, it also assumes that the region of space under analysis is inside the overlapping field of view (FOV) of all cameras. If a scene point is outside the FOV of one or more cameras, the missed detection causes the remaining, possibly correct detections from other views to be discarded. This obvious problem is corrected by modifying the fusion operator as:

$$\log(P(X \mid x_1, x_2 \ldots, x_n)) \propto (\Sigma_i \delta(x_i)) \sum_{i=1}^{n} \frac{1}{\tau C_i} \Gamma(x_i), \tag{3.7}$$

where $\delta(x) = \begin{cases} 1 & \text{if } x \text{ inside image dimensions;} \\ 0 & \text{otherwise.} \end{cases}$

and $\Gamma(x) = \begin{cases} \log(L(x)) & \text{if } \delta(x) = 1; \\ 0 & \text{otherwise.} \end{cases}$

The form of equation 3.7 ensures that if a scene point is outside the field of view of a camera, that particular view will not effect the fusion results. Alse the normalizing term $\Sigma_i \delta(x_i)$ guarantees higher confidence in regions with greater view overlap.

A pixel p, in a reference view can be classified as an image of the reference scene plane localization of an object if the occupancy likelihood given by equation 3.7 is above a threshold. In the case foreground objects are people and the reference scene plane is the ground plane, pixel p will correspond to the feet of a person in the scene. Since pixel p and its warped locations in other views $p'_1, p'_2 \ldots, p'_n$ all have the same piercing point, they all correspond to the same location on the reference scene plane. Therefore by finding p in the reference

view that satisfies the HOC, we have in fact, localized the particular person in all views (i.e $p'_1, p'_2 \ldots, p'_n$). This strategy implicitly resolves the issue of correspondences across views and makes the choice of reference view irrelevant (chosen arbitrarily).

3.3.2 Localization Algorithm

Our algorithm for locating people is quite straight forward. First we obtain the foreground likelihood maps in each view. This is done by modelling the background using a mixture of gaussians [127][46] and finding the probability for each pixel belonging to the foreground. In the second step instead of warping every pixel in the reference image to every other view we perform the equivalent step of warping the foreground likelihood maps from all the other views on to the reference view. These warped foreground likelihood maps are then fused according to equation 3.7 to produce what we call a 2D grid of reference plane occupancy likelihoods. Following are the steps in our algorithm that are also shown in figure 3.4:

Objective Localize people on a reference plane.

1. Obtain the foreground likelihood maps $\Psi_1, \Psi_2 \ldots, \Psi_n$.
    - Model Background using Mixture of Gaussians.
    - Perform Background Subtraction to obtain foreground likelihood information.
2. Obtain reference plane homographies.
3. Warp foreground likelihood maps to a reference view using homographies of the reference plane.

- Warped Foreground Likelihood maps: $\Psi'_1, \Psi'_2 \ldots, \Psi'_n$

4. Fuse $\Psi'_1, \Psi'_2 \ldots, \Psi'_n$ at each pixel location p of the reference view according equation 3.7 to obtain synergy map θ.

3.3.3 Tracking

Our tracking methodology is based on the concept of spatio-temporal coherency of scene occupancies created by objects. Assuming that a particular scene location at a specific time can be occupied by only a single individual, we hypothesize that over time spatially coherent scene occupancies correspond to the tracks of scene objects. We therefore propose a look-ahead technique to solve the tracking problem using a sliding window over multiple frames. This information gathering over time, of systems simulating the cognitive processes is supported by many researchers in both computer vision and psychology (e.g., [58], [59], [108]). Neisser [108] proposed a model according to which the perceptual processes continually interact with the incoming information to verify hypotheses formed on the basis of available information up to a given time instant. Marrs principle of least commitment [59] states that any inference in a cognitive process must be delayed as much as possible. Many existing algorithms use similar look-ahead strategies or information gathering over longer intervals of time (for example, by backtracking) [109] [110].

Let us denote by $\xi^{\mathbf{n}} = (\xi_1^n, \xi_2^n, ..., \xi_t^n)$ the trajectory of spatio-temporal occupancies by individual n, where ξ_i^n, represents the spatial localization of the individual n in the occupancy likelihood information θ_i at time i. Given the occupancy likelihood information from

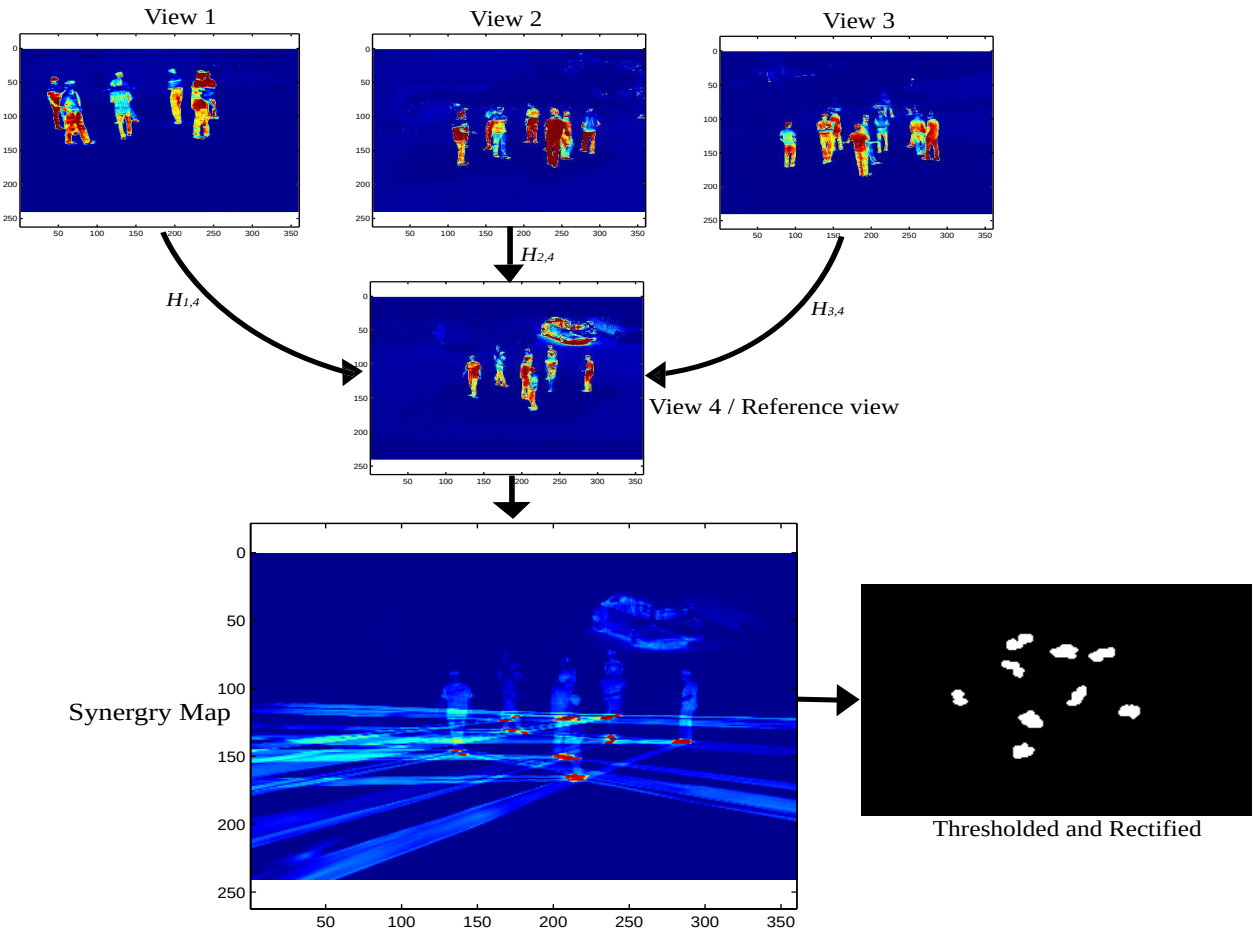


Figure 3.4: The four smaller images are foreground likelihood maps obtained from the background model (mixture of gaussians) on the images shown in figure 1. In all images in the figure the colormap used assigns a hotter palette to higher values. View 4 was chosen as the reference view. The image on the bottom is the synergy map obtained by warping views 1, 2, and 3 onto view 4 and multiplying them together. The pixels representing the ground locations of the people are segmented out by applying an appropriate threshold. The binary image shown is the result of applying the threshold and rectifying with the ground plane (the white regions corresponding to the feet).

our localization algorithm for a sliding time window of t frames $\theta_1, \theta_2, ..., \theta_t$, the tracks are obtained by maximizing the posterior conditional probability:

$$[\hat{\xi^{\mathbf{1}}}, ..., \hat{\xi^{\mathbf{n}}}] = \arg\max_{l_1,...,l_n} P(\xi^{\mathbf{1}} = l_1, ..., \xi^{\mathbf{n}} = l_n | \theta_1, \theta_2, ..., \theta_t). \tag{3.8}$$

To achieve this we define and energy function that combines occupancy regularization and region information, in a fashion similar to Mumford-Shah style functions. The global minimum is found by using graph cut techniques that is discussed next.

3.3.4 Trajectory Segmentation Using Graph Cuts

For a time window of t frames, we obtain the scene occupancy likelihood information from our localization algorithm: $\theta_1, \theta_2, ..., \theta_t$. Each θ_i is a 2D grid of object occupancy likelihoods, obtained from multi-view fusion at multiple scene planes as described in previous sections. Now by arranging Θ_is in the time dimension we create what we call a three dimensional *spatio-temporal occupancy likelihood grid*: $\mathbf{\Theta} = [\theta_1; \theta_2; ...; \theta_t]$. Each location or node in this 3D grid contains the object presence likelihood for a specific space-time point. Our goal is to segment $\mathbf{\Theta}$ into background (non-occupancies) and object occupancy trajectories with the following criteria:

1. Grid locations with high occupancy likelihoods have higher chance of being included in object trajectories.

2. Object trajectories be spatially and temporally coherent.

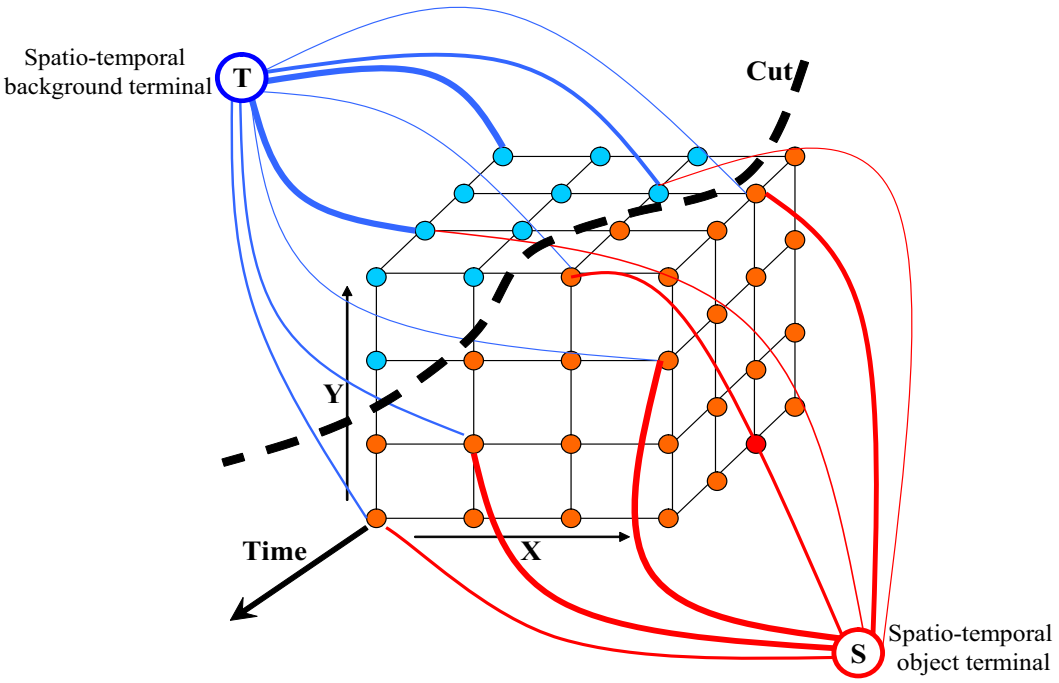


Figure 3.5: Spatio-temporal grid of object occupancy likelihoods. The 3D grid is Θ at a particular time instant (each XY plane in the grid is a synergy map consisting of object occupancy likelihoods). The segmentation of coherent spatio-temporal occupancies delivers the tracks. This is done with graph cuts.

Given these criteria we define our energy function as:

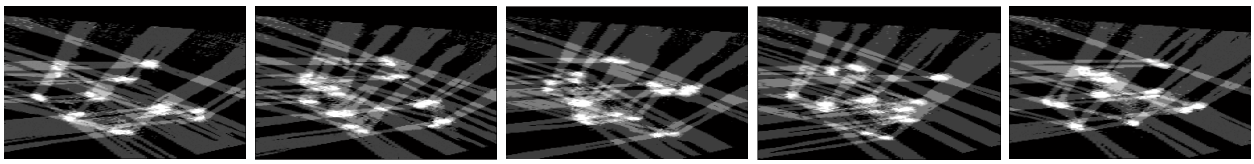
$$\mathcal{E} = \mu \sum_{p \in \mathcal{P}} -\hat{\mathbf{\Theta}}(p) + \sum_{(p,q) \in N} B_{p,q}, \tag{3.9}$$

where $\mathcal{P}$ is the set of all grid locations/nodes in $\mathbf{\Theta}$, N is the set of grid locations in a neighborhood, and,

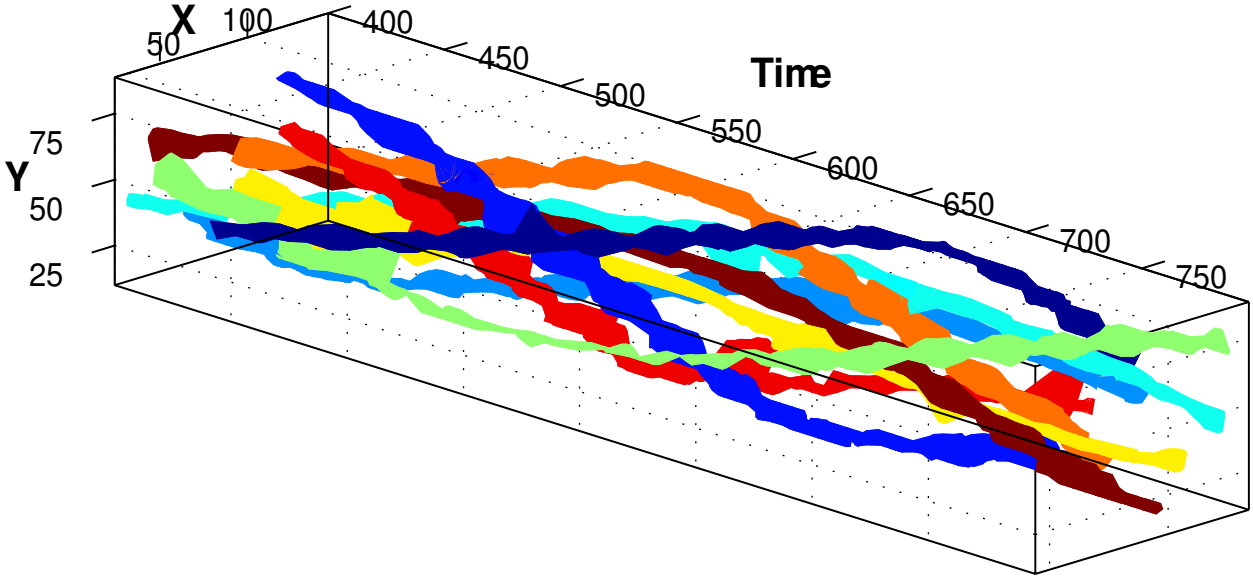
$B_{p,q} \propto e^{-dist(p,q)/2\tau^2}$, where $dist(p,q)$ is the 3D Euclidian distance between grid locations p and q and τ is a normalizing factor. The first term in equation 3.9, also known as the data term, imposes scene occupancy. The second term known as the smoothness term imposes the constraint of spatio-temporal coherency. By minimizing equation 3.9, the idea is to obtain regions in the spatio-temporal occupancy likelihood space that have high presence probabilities (small negative log likelihood) and are smooth i.e. close to each other both in space and time.

In order to minimize the energy function given in equation 3.9, we use graph cut techniques. Our undirected graph $G = (V, E)$ is as follows. The set of vertices is the set of spatio-temporal grid locations augmented by the source S and sink T vertices: $V = \mathcal{P} \cup S, T$. The set of edges consists of all neighboring pairs of nodes, along with an edge between each node and the source and sink: $E = \mathcal{N} \cup \{(p, S), (p, T) : p \in \mathcal{P}\}$. In terms of the weights on the edges, there are three cases to consider. If $(p, q) \in \mathcal{N}$ then $w(p, q) = B_{p,q}$.

On the other hand, if the edge contains the source S or sink T as one of its vertices, then $w(p, \{S, T\}) = -\mathbf{\Theta}(p)$. It is relatively straightforward to show that the minimum cut on the graph G corresponds to the minimum values of the energy function equation 3.9



(a)



(b)

Figure 3.6: Figure (a) shows a sequence of synergy maps at the ground reference plane of 9 people obtained using our algorithm. In (b) we show the segmented tracks of the people using our approach. Different tracks are colored differently to help in visualization. The spiralling pattern of the worms is only a coincidence. This resulted because the people were walking in circles in this particular sequence.

[111]. The specific algorithm we use is the α-expansion algorithm described in [112]. To keep the problem computationally tractable we quantized the XY plane as a 100x100 grid. The sliding time window size was kept at 15 frames for each experiment (which corresponds to 1 second in the real world of a 15fps video). The sliding window has minimal overlap, 1 frame, e.g. last frame of window$_i$ and first frame of window$_{i+1}$ are the same. The overlap is used to pass on the track identities. The identities are initialized in the first window. For successive windows each segmented track is given the ID that a previous window assigned it at the overlap frame. Though larger window size and greater overlap can be used to improve performance, we found the improvement was not significant enough to justify the increased load of processing.

Note that we do not make hard detection decisions and use them for tracking. Rather, tracking and detection are intimately tied together and are performed simultaneously when we segment out space-time tracks from the occupancy likelihood data $\mathbf{\Theta}$. This, we believe is an elegant solution to the inherently coupled tasks of detection and tracking. The advantage of this approach is twofold. First, false negatives and false positives are reduced compared with a traditional threshold based detection (see experimental evaluation). In cases where a missed detection from thresholding (for one or more frames) would cause a track to be lost, we are able to recover the tracks, and hence the detections. This is because the energy functional minimized with graph cuts combines both occupancy probability and spatio-temporal smoothness. For instance, if the occupancy probability for a person does not pass the detection threshold for a particular frame in the time window, our track segmentation

approach still includes the region in the particular frame to reduce the cost incurred by having neighboring nodes in the space-time occupancy grid more than one frame away (smoothness). Second, this approach helps in cases where a thresholded detection results in artifacts in a single person's detection i.e. the region is split into two or more very close but unconnected regions. Such regions are typically merged together by our approach. This property of our approach may also cause tracks of two or more people to merge if they come very close to eachother; however these are uncommon cases and resolved in the long run, since people's body parts tend to remain closer to them than to other people. In situations where detection results using a thresholding approach are sufficiently good (typically not the case in challenging scenarios as demonstrated in our results section), a simple tracker like EKF or the more sophisticated particle filtering tracking can be used as has been attempted in past literature [6] [12]. But such trackers don't naturally handle splits and merges, and require an explicit split-merge analysis separate from the tracking. This of course is naturally handled in our track segmentation approach. In figure 3.6(b) we describe an example of the spatio-temporal occupancy tracks obtained for a scene containing 9 people, which was used in one of our experiments. The figure shows the results after processing multiple time windows.

3.4 Experimental Results

This scene was captured using a video-surveillance dedicated setup of 4 synchronized cameras in a parking lot. The cameras were mounted at various heights ranging from 2m$\sim$6m and arranged unevenly in a rough circle. The sequence is over 3000 frames and contains between

5∼9 people. The people were constrained to move in an area of approximately 5 meters by 5 meters to simulate dense crowds and severe occlusions. We were attempting to increase the density of people and vary the number of views, in order to study the breakdown thresholds and other characteristics.

Figure 3.7 shows our tracking results on this dataset. Due to the density of the gathering, occlusions were abundant and quite severe. An interesting thing to note is the color similarity of the people in the scene. A method that uses appearance (color distribution) matching across views would perform poorly in such a situation, whereas our method performs quite well. The top row of figure 3.7 shows a visualization of the top view, with configuration of the cameras overlaid. Camera overlap is color coded so that brighter yellow corresponds to higher overlap. The blue and red squares in the top view depict the true and false positives respectively.

In table 1 and figure 3.8 we show the quantitative and qualitative analysis of our results on this sequence. We analyzed the accuracy of our tracking results by comparing them with the manually marked ground truth. Our tracking accuracy measure was the Euclidean distance between the ground truth and the tracking localization. For this sequence we had access to metric calibration data in order to convert image distances to actual world distances in terms of inches. The error was calculated for each tracked person in each view and was averaged over the number of people and the number of views. False positives and negatives were *not* included in the calculation of this measure. We call this the *total average track error*. Figure 3.8(a) is the plot of the total average track error, computed at intervals of 100

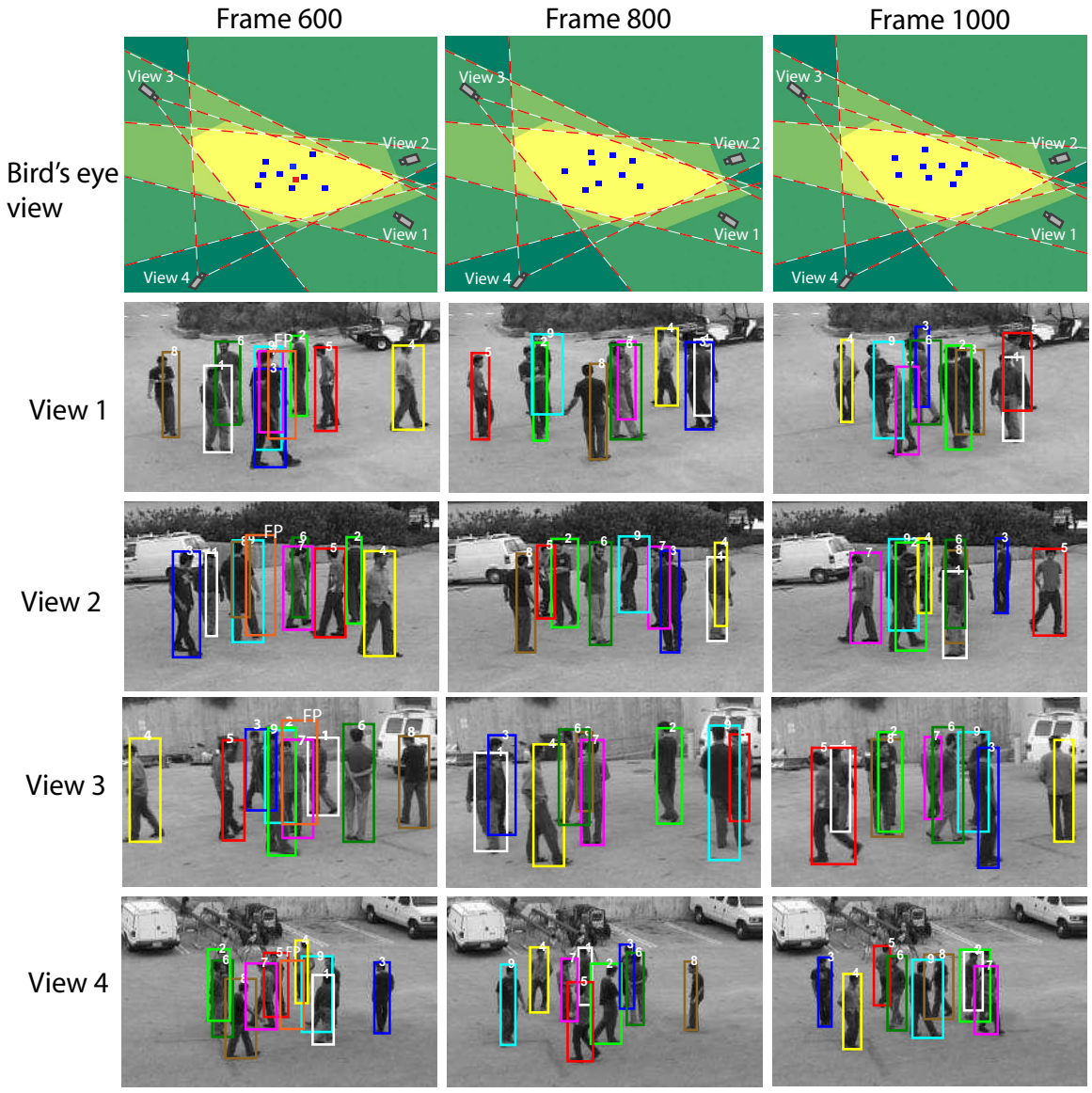


Figure 3.7: *Parking Lot Sequence:* Tracking results for a scene containing 9 people captured from 4 view points. The first row shows a visualization of the top view. It shows the camera field of view overlap, with higher overlap corresponding to yellower regions. Detection true and false positives are shown with blue and red squares, respectively. Rows 2-5 show the four camera views of the scene. Left to right, the columns correspond to frames 600, 800 and 1000 in the respective views.

	Person 1		Person 2		Person 3		Person 4		Person 5		Person 6		Person 7		Person 8		Person 9	
	4 v	2 v	4 v	2 v	4 v	2 v	4 v	2 v	4 v	2 v	4 v	2 v	4 v	2 v	4 v	2 v	4 v	2 v
Frame 300	5.39	7.46	3.10	7.08	1.64	2.56	2.54	11.6	3.47	5.24	1.59	17.4	7.14	21.6	9.35	18.1	--	--
Frame 350	9.73	5.05	1.46	24.1	5.19	9.76	3.13	3.00	0.81	7.55	2.77	--	2.06	16.5	3.17	9.32	--	--
Frame 400	8.04	7.43	3.6	4.49	11.6	12.4	6.20	5.57	3.01	2.64	2.56	12.9	0.45	8.35	4.17	10.9	--	--
Frame 450	2.66	6.41	2.25	32.5	0.40	7.56	0.66	4.65	2.19	16.5	2.61	7.7	10.8	7.91	15.0	10.5	11.3	6.66
Frame 500	5.3	--	7.66	18.9	3.95	20.0	3.06	3.27	5.0	22.3	3.38	15.5	5.51	19.6	5.48	18.1	0.64	19.8
Frame 550	3.48	7.81	1.31	32.5	1.55	21.5	5.17	4.95	3.79	15.2	2.32	15.1	2.5	--	3.45	0.85	1.8	--
Frame 600	3.3	3.4	13.9	10.6	3.33	20.7	11.2	10.0	25	--	7.8	24.2	3.54	4.82	4.3	16.2	0.45	22.4
Frame 650	0.66	4.20	2.19	2.44	11.6	5.75	6.20	20.1	3.01	--	4.85	21.7	3.07	13.0	4.93	22.7	5.9	10.2
Frame 700	3.01	2.78	2.56	5.50	0.45	11.4	1.46	32.5	5.19	22.9	3.13	19.7	0.81	16.3	5.28	6.12	6.2	5.46
Frame 750	2.3	2.97	11.5	4.41	2.36	6.86	5.6	8.93	4.56	3.54	9.8	31.6	3.5	2.78	4.1	18.6	3.3	7.13
Frame 800	4.9	5.75	1.86	15.1	2.15	22.2	3.75	30.5	3.2	17.2	4.57	--	3.86	8.2	4.35	23.8	1.2	23.6
Frame 850	2.81	12.7	5.5	9.72	3.0	--	1.04	19.3	1.82	--	0.75	13.5	5.7	1.87	14.5	20.3	1.45	21.9
Frame 900	5.5	27.1	3.69	10.8	6.75	--	3.45	21.6	2.98	8.26	2.65	15.0	3.46	3.24	13.2	3.52	2.35	10.1
Frame 950	4.12	27.4	4.48	--	4.6	11.8	9.9	28.0	4.6	14.0	5.4	5.82	12.4	28.6	11.5	17.2	2.74	12.5
Frame 1000	5.6	10.3	4.56	21.5	1.86	13.5	2.15	6.75	3.48	19.2	1.31	4.32	1.55	2.66	8.55	2.01	3.5	20.7

Table 3.1: Track Error from Ground Truth for Parking Lot Dataset (distance in inches)

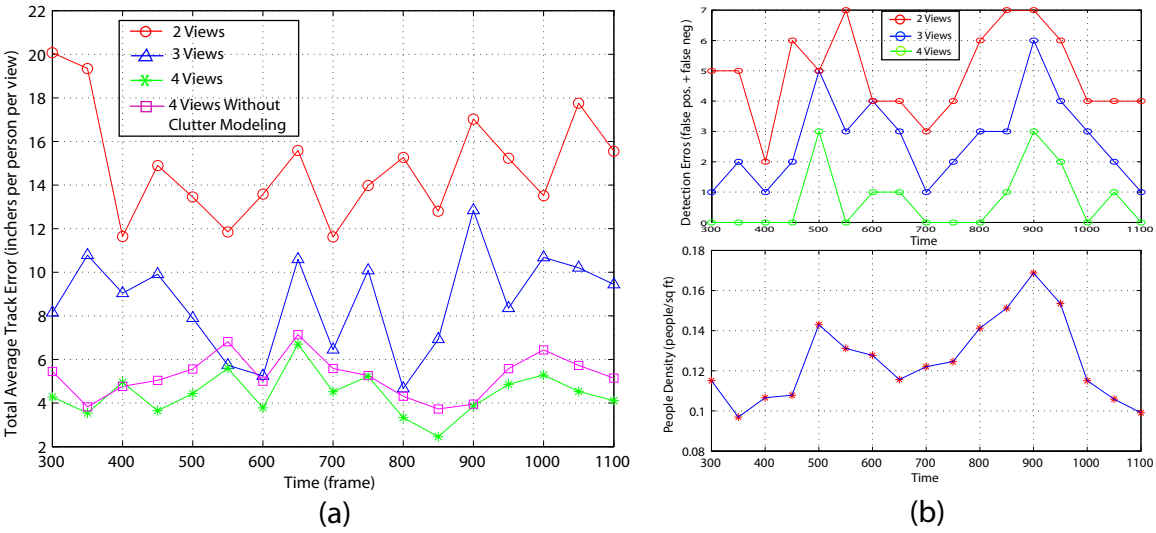


Figure 3.8: *Parking Lot Dataset:* (a)Total average track error of persons tracked over time. Pixel distances were converted to inches in the scene using metric calibration data. As expected track error increases with lesser number of view. This is essentially because of imprecise localization. Also the track error increases if clutter is not modelled as can be seen for the magenta plot. (b) Plot on the top shows the detection error (number of false positives + number false negatives) over time. The bottom plot shows the variation of the people density over time. Notice the correlation between detection error and people density. As can be seen increasing density effects the performance of our algorithm. Higher density means more inter-person occlusions for any vantage point and thus more detection errors.

frames. We varied the number of views by selecting a subset of the available views, in order to study the effect of reducing views on our approach. As expected, the total average track error significantly increased, from a mean of approximately 4 inches with four views (green plot) to over 14 inches with two views (red plot). The magenta plot in the figure shows the track error with 4 views, if clutter modelling is not used. As shown, the accuracy of tracking decreases if clutter modelling is not done. Clutter modelling helped in making the tracks more streamlined and precise by effectively pruning out false occupancy information in the periphery of detections. Also in some cases (frames 550, 600, 800) the track error with 3 views and clutter modelling (blue plot) is close to that of 4 views without clutter modelling. Although we do not expect this particular trend to be the general pattern, it does indicate that modelling clutter has a useful impact. Detection error on the other hand was relatively unaffected with the use of clutter modelling e.g. false detections arising due to scene locations being occluded from every view are not affected by using clutter modelling.

In our opinion the other most significant factor influencing the performance of our algorithm is the density of the crowd or gathering. The greater the density the more scene occupancies per unit area and therefore greater occlusions from vantage points resulting in difficulty with detection and localization. In figure 3.8(b) we show three plots depicting a correlation between the density of people in the scene and the resulting detection error (sum of false positives and missed detections). To obtain people density we calculate the area of the convex hull (in sq. feet, recall that we have metric calibration) containing the ground plane localizations of all tracked people at a given time instant and divided the number of

people by this value. The people density varied from 0.09∼0.17 persons/sq feet as can be seen in the bottom plot of figure 3.8(b). The top plot shows the detection error versus time at intervals of 100 frames. Notice the correlation, especially at the peaks of people densities. The correlation coefficient between the density plot and detection error plot is 0.7 for four views, 0.75 for three views and 0.62 for 2 views.

Although we acknowledge that the number of views and people density are not the only scene factors influencing the performance of our approach, we believe that these are the most crucial. A detailed analysis can be quite exhaustive, and can include camera configuration, relative people configuration (certain formations of people can occlude scene regions from all cameras) and scene geometry. These and other factors are beyond the scope of this paper and will be addressed in future studies.

We have also experimented with utilizing a simple threshold based detection approach, to empirically test the advantage of using our graph cuts track segmentation approach. We empirically set the most optimal threshold (running several times and selecting the best threshold) on occupancy likelihoods $\mathbf{\Theta}$ obtained from the localization algorithm. At each frame we have regions detected as people. We obtain the connected components on these regions and put a minimum size threshold on these to prune out the noise. Detection error is then computed as specified earlier. Plots of the detection error from the simple thresholding approach are shown in figure 3.9 and can be compared with our approach. As shown by the data, simple thresholding results in many more detection errors compared with our approach, thus corroborating our claim that the integration of detection and tracking in a unified

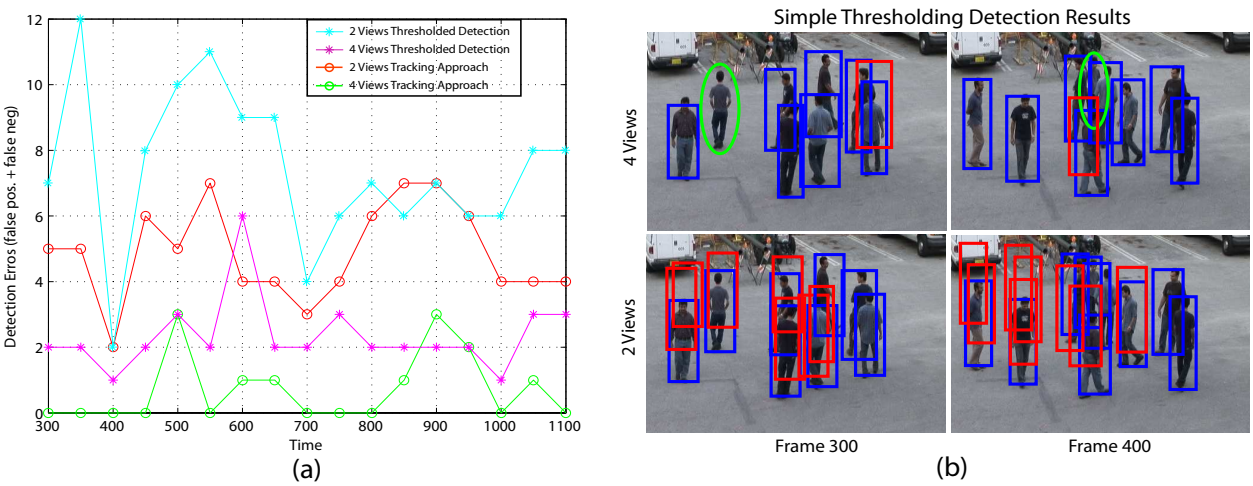


Figure 3.9: *Parking Lot Dataset:* (a)Detection error for utilizing a simple thresholding of the occupancy likelihood data compared with our trajectory segmentation based approach. Plots for detection error using 2 views and 4 views are shown. As can be clearly seen our approach performs much better. (b) Detection results using a threshold based approach. Blue rectangles are true positives, red rectangles are false positives and green ellipses are false negatives.

track segmentation formulation is desirable. Though it may be argued that threshold based detection results can be improved by incorporating more sophisticated models, like human shape priors, we maintain that such models can be used to augment our track segmentation approach (see [72] for a use of shape priors in graph cuts based segmentation).

3.5 Summary

In this chapter we have presented an algorithm that can reliably track multiple people in a complex environment. This is achieved by resolving occlusions and localizing people on a reference scene plane (typically the ground plane) using a planar homographic occupancy constraint. Combining foreground likelihood information from multiple views and obtaining the global optimum of space-time scene occupancies over a window of frames we segment out the individual trajectories of the people. We have presented detailed quantitative results on a challenging multi-view dataset.

As mentioned earlier, the HOC is not limited to any particular plane in the scene. In fact as we discuss in the next chapter, the HOC can be extended to multiple scene planes for greater robustness in tracking. But more crucially in the next chapter we develop the link between the HOC and image-based visual hull intersection that is used to generate the 3D structure of objects from 2D views.

CHAPTER 4
HOMOGRAPHIC OCCUPANCY CONSTRAINT AND VISUAL HULL INTERSECTION ON MULTIPLE SCENE PLANES

4.1 Introduction

In this chapter we provide a link between the homographic occupancy constraint (HOC) presented above and visual hull intersection leading to a purely image-based approach for 3D reconstruction [2]. The HOC is shown to be applicable on multiple scene planes. Each planar homographic fusion delivers a cross-sectional slice of the object cut out by the plane used for the HOC, which are accumulated to obtain object structure. The approach is also used to increase the robustness of our localization and tracking algorithm.

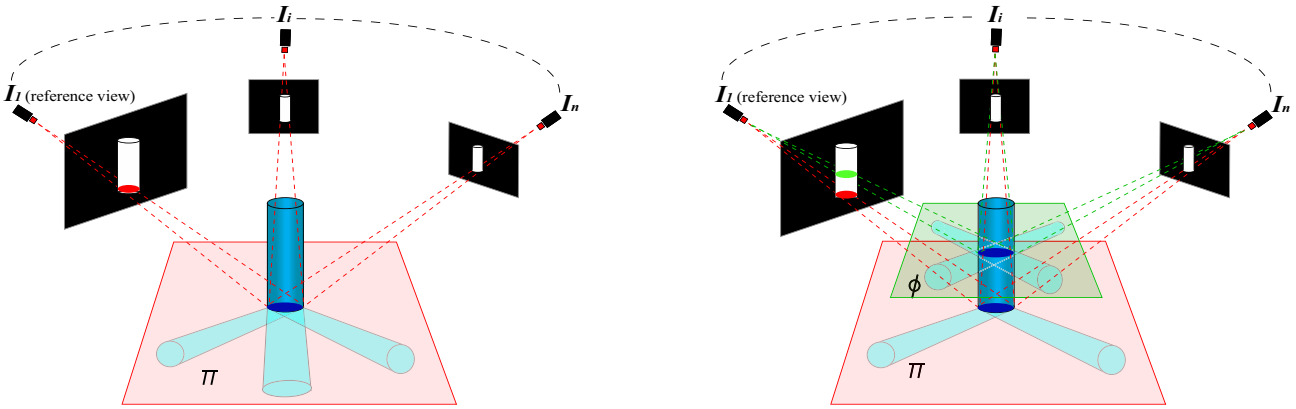
4.2 Obtaining Object Slices

Consider figure 4.1 (a). The scene is viewed from several angles with the cylinder object detected as foreground (white regions) in each view. Let us apply the HOC on this scene. One of the views, say I_1, is chosen as the reference view. Warping view I_i to the reference view using homography $H_{i_\pi 1}$ induced by scene plane π, first every foreground pixel in I_i is projected to its piercing point on π. This process can be viewed as the foreground object casting a *shadow* on π (an analogy if the cameras are replaced by point light sources), as

depicted by the light blue regions in figure 4.1(a). The shadow is then projected onto the reference view to complete the operation of the homographic warping.

Clearly computing the shadow is equivalent to determining the region on π that falls inside the visual hull of the object image in I_i. The fusion of these shadows projected from various views therefore amounts to performing visual hull intersection on plane π, depicted by the dark blue region in figure 4.1(a). This process is performed implicitly when we apply the HOC i.e. warp all the views onto the reference view and fuse them to obtain the red region in the reference view I_1. Without loss of generality, reference image plane I_1 after homographic fusion of foreground data can be viewed as a projectively transformed planar slice of the object (strictly speaking a perspectivity with only 6dof).

In our implementation as discussed in the previous chapter, instead of using binary foreground maps, we pursue a more statistical approach and model the background [127] in each view to obtain foreground likelihood maps, thereby using cameras as statistical occupancy sensors (foreground interpreted as occupancy in space). In the case of non-stationary cameras object detection is achieved in a plane+parallax framework [51] assigning high foreground likelihood where there is high motion parallax. The reason to adopt a $soft$ approach is to delay the act of thresholding preventing any premature decisions on pixel labelling; an approach that has proven to be very useful in visual hull methods [130] due to their susceptibility to segmentation and calibration errors. Let us restate I_i as the foreground likelihood map (each pixel value is likelihood of being foreground) in view i of n. Consider a



(a) (b)

Figure 4.1: Warping the silhouettes of an object from the image plane to a plane in the scene using a planar homography is equivalent to projecting the visual hull of the object onto the plane. If the camera center is considered as a point light source this can be interpreted as the object casting its shadow on a plane. Figure (a) demonstrates this for a cylinder viewed from different angles. The intersection of these *shadows* amounts to performing visual hull intersection on the plane. The result is the dark blue region that can be considered a slice of the cylinder cut out by π. This process is implicitly performed when we warp and fuse silhouette information from other views on to reference view I_1 and is depicted by the red region. (For the sake of clarity projection of the shadows are not shown in the reference view, and only the intersection of these projections i.e. the red region is shown). Figure (b) demonstrates that the same process can be performed on a second plane ϕ delivering another slice of the cylinder.

reference plane π in the scene inducing homographies $H_{i_\pi j}$ from view i to view j. Warping I_i's to a reference view I_{ref} we have the warped foreground likelihood maps: $\hat{I}_i = [H_{i_\pi ref}]I_i$.

Visual hull intersection on π by the homographic fusion of the foreground likelihood maps:

$$\phi_{ref} = \prod_{i=1}^{n} \hat{I}_i, \tag{4.1}$$

where ϕ_{ref} is the projectively transformed grid of object occupancy likelihoods. Arguably a more elaborate fusion model can be used at the expense of simplicity but that is not the primary focus of this research. Indeed a sensor fusion strategy that explicitly models pixel visibility, sensor reliability, scene radiance as in [84], can be transparently incorporated without affecting our underlying approach of fusing at slices in the image plane rather than in 3D space.

Each value in ϕ_{ref} is saying what the likelihood is of this grid location being inside the body of the object; indeed representing a slice of the object cut out by plane π. It should be noted that the choice of reference view is irrelevant as the slices obtained on all image planes and the scene plane π are projectively equivalent. This computation can be performed at an arbitrary number of planes in the scene, each giving a new slice of the object. Naturally this does not apply to planes that do not pass through the object's body since visual hull intersection on these planes will be empty, therefore a separate check is not necessary. Figure 4.1(b) demonstrates a second slice of the cylinder obtained using our approach.

Starting with a reference plane in the scene (typically the ground plane) we perform visual hull intersection on successively parallel planes in the up direction along the body of

the object. The probabilistic occupancy grids ϕ_is obtained in this fashion can be thresholded to obtain object slices, but this creates the problem of finding the optimum threshold at each slice level. Moreover, the slices have a strong dependency on each other as they are parts of the same object/s and should as such be treated as a whole. Our approach is to model this dependency by stacking up the slices, creating a three dimensional data structure $\Phi = [\phi_1; \phi_2; \ldots \phi n]$. Φ is not an entity in the 3D world or a collection of voxels. It is simply put, a logical arrangement of planar slices, representing discrete samplings of the continuous occupancy space. Object structure is then segmented out from Φ i.e., simultaneously from all the slices as a smooth surface that divides the space into the object and background. Details of this process are delayed until section 4.3. In the next section we present an image-based approach using the homography of a reference plane in the scene to compute homographies induced between views by planes parallel to the reference plane.

4.2.1 Extending to Successive Planes

Consider a coordinate system XYZ in space. Let the origin of the coordinate frame lie on the reference plane, with the X and Y -axes spanning the plane. The Z-axis is the reference direction, which is thus any direction not parallel to the plane. The image coordinate system is the usual xy affine image frame, and a point $\mathbf{X}$ in space is projected to the image point $\mathbf{x}$ via a 3×4 projection matrix $\mathbf{M}$ as:

$$\mathbf{x} = \mathbf{MX} = [m_1 \quad m_2 \quad m_3 \quad m_4]\mathbf{X},$$

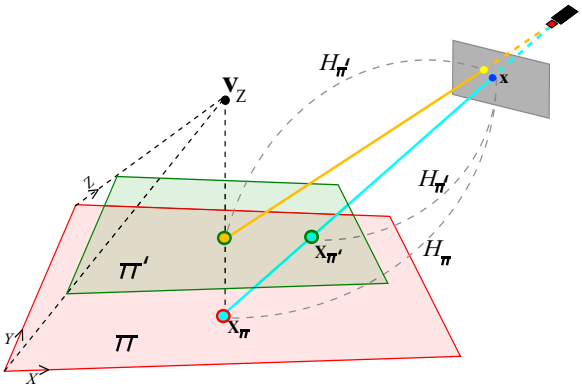


Figure 4.2: The diagram illustrates the geometrical relationship of the homography of an image plane to two parallel scene planes π and π'. $\mathbf{v}_Z$ is the vanishing point of the direction normal to π and π'. Given the homography H_π from the image plane to π, $H_{\pi'}$ can be computed by adding a scalar multiple of the vanishing point $\mathbf{v}$ to the last of column of H_π.

where $\mathbf{x}$ and $\mathbf{X}$ are homogenous vectors in the form: $\mathbf{x} = (x, y, w)^T$, $\mathrm{X} = (X, Y, Z, W)^T$, and '=' means equality up to scale. The projection matrix $\mathbf{M}$ can be parameterized as:

$$\mathbf{M} = [\mathbf{v}_X \quad \mathbf{v}_Y \quad \mathbf{v}_Z \quad \hat{\mathbf{l}}],$$

where $\mathbf{v}_X$, $\mathbf{v}_Y$ and $\mathbf{v}_Z$ are the vanishing points for X, Y and Z directions respectively and $\hat{\mathbf{l}}$ is the vanishing line of the reference plane normalized [53].

Suppose the world coordinate system is translated from the plane π onto the plane π' along the reference direction(Z) by z units as shown in figure 4.2, then the new projection matrix $\mathbf{M}'$ is parameterized as:

$$\mathbf{M}' = [\mathbf{v}_X \quad \mathbf{v}_Y \quad \mathbf{v}_Z \quad \alpha z \mathbf{v}_Z + \hat{\mathbf{l}}],$$

where α is a scale factor. Columns 1, 2 and 4 of the projection matrices are the three columns of the respective plane to image homographies. Therefore, the plane to image homographies can be extracted from the projection matrices, ignoring the third column, to give:

$$H_\pi = [\mathbf{v}_X \quad \mathbf{v}_Y \quad \hat{\mathbf{l}}], \quad H'_\pi = [\mathbf{v}_X \quad \mathbf{v}_Y \quad \alpha z \mathbf{v}_Z + \hat{\mathbf{l}}].$$

In general:

$$H_\gamma = H_{ref} + [0|\gamma \mathbf{v}_{ref}], \tag{4.2}$$

where H_{ref} is the homography of the reference plane γ is a scalar multiple encapsulating α and z, [0] is a 3x2 matrix of zeros and $\mathbf{v}_{ref}$ is the vanishing point of the reference direction. Let $H_{i_\pi j}$ be the homography between views i and j induced by scene plane π. Now $H_{i_\pi j}$ can be decomposed as the product of two homographies first from i to π and then from π to j:

$$H_{i_\pi j} = [H_{\pi_{to} j}][H_{i_{to}\pi}]. \tag{4.3}$$

Similarly the homography $H_{i_\phi j}$ induced by a plane ϕ that is parallel to π can be written as:

$$H_{i_\phi j} = [H_{\phi_{to} j}][H_{i_{to}\phi}]. \tag{4.4}$$

Now from equation 4.2 we have:

$$H_{\phi_{to} j} = [H_{\pi_{to} j}] + [0|\gamma \mathbf{v}_{ref}]. \tag{4.5}$$

$$H_{i_{to}\phi} = inv(H_{\phi_{to} i}) = inv([H_{\pi_{to} i}] + [0|\gamma \mathbf{v}_{ref}]) = H_{i_{to}\pi} - \frac{1}{1+g}[H_{i_{to}\pi}][0|\gamma \mathbf{v}_{ref}][H_{i_{to}\pi}], \tag{4.6}$$

where $g = trace([0|\gamma \mathbf{v}_{ref}][H_{i_{to}\pi}])$. Replacing 4.5 and 4.6 into 4.4 we have:

$$H_{i_\phi j} = (H_{\pi_{to} j} + [0|\gamma \mathbf{v}_{ref}])(H_{i_{to}\pi} - \frac{1}{1+g}[H_{i_{to}\pi}][0|\gamma \mathbf{v}_{ref}][H_{i_{to}\pi}].) \tag{4.7}$$

Since $H_{i_{to}\pi}$ is a central projection from one plane to another (2D perspectivity with 6 DOF) the last row is [0 0 1]; therefore, $g = trace([0|\gamma \mathbf{v}_{ref}][H_{i_{to}\pi}]) = \gamma$. Plugging this and 4.3 into 4.7 and with some matrix algebra we reach:

$$H_{i_\phi j} = (H_{i_\pi j} + [0|\gamma \mathbf{v}_{ref}])(I_{3\times 3} - \frac{1}{1+\gamma}[0|\gamma \mathbf{v}_{ref}]). \tag{4.8}$$

This result shows that if we have the homography $H_{i_\pi j}$ induced by a reference scene plane π between views i and j, and the vanishing point of a reference direction then the homography $H_{i_\phi j}$ induced by a plane ϕ parallel to π in the reference direction is given by can be obtained easily with some matrix algebra.

In our implementation we used the ground plane as the reference scene plane and the up direction as the reference direction. The ground plane homographies between views were automatically calculated with SIFT [54] feature matches and using the RANSAC algorithm [55]. Vanishing points for the reference direction were computed by detecting vertical line segments in the scene and finding their intersection in a RANSAC framework as in [56]. It should be noted that the particular values of γ are not significant, we are only interested in the range of γ for planes that span the body of the object (e.g., if the object is a person, then starting from the ground plane to a plane parallel to the ground plane but touching the tip of the head). The computation of this range for γ is quite straightforward since outside this range visual hull intersection on the corresponding planes will be empty. In the next section we describe how we segment out the object from the occupancy grid data.

4.3 Object Segmentation

As described earlier slices computed along the body of the object are stacked, creating a three dimensional data structure Φ that encapsulates the object structure. To segment out the object we evolve a parameterized surface $\mathcal{S}(q) : [0, 1] \rightarrow \mathbb{R}^3$, that divides Φ between the object and the background similar to the approach in [119]. This is achieved by formulating the problem in a variational framework, where the solution is a minimizer of a global cost functional that combines a smoothness prior on slice contours and a data fitness score. Our energy functional is defined as:

$$E(\mathcal{S}) = \int_{\mathcal{S}} g\left(|\nabla\Phi(\mathcal{S}(q))|\right)^2 dq + \int_{\mathcal{S}} \left|\frac{\partial\mathcal{S}(q)}{\partial q}\right|^2 dq, \tag{4.9}$$

where $\nabla\Phi$ denotes gradient of Φ, and g denotes a strictly decreasing function: $g(x) = 1/(1 + x^2)$. The first term at the right side of (4.9) represents external energy. Its role is to attract the surface towards the object boundary in Φ. The second term, called the internal energy computes, the area of the surface. Given the same volume, smoother surface will have smaller area. Therefore, this term controls the smoothness of the surface to be determined. When the overall energy is minimized, the object boundary will be approached by a smooth surface.

Minimizing energy functional (4.9) is equivalent to computing geodesic in a Riemannian space:

$$E(\mathcal{S}) = \int g\left(|\nabla\Phi(\mathcal{S})|\right) \left|\frac{\partial\mathcal{S}}{\partial q}\right| dq. \tag{4.10}$$

With the Euler-Lagrange equation deduced, this objective function can be minimized by using the gradient descent method by an iteration time t as

$$\vec{\mathcal{S}}_t = g\left(|\nabla\Phi(\mathcal{S})|\right)\kappa\vec{\mathcal{N}} - \left(\nabla g\left(|\nabla\Phi(\mathcal{S})|\right)\cdot\vec{\mathcal{N}}\right)\vec{\mathcal{N}}, \tag{4.11}$$

where κ is the surface curvature, and $\vec{\mathcal{N}}$ is the unit normal vector of the surface.

Since the objects to be reconstructed may have arbitrary shape and/or topology as shown in our experiments, the segmentation is implemented using the level set framework [48]. Level sets based methods allow for topological changes to occur without any additional computational complexity, because an implicit representation of the evolving surface is used. The solution (4.11) can be readily cast into level set framework by embedding the surface $\mathcal{S}$ into a 3D level set function Ψ with the same size as Φ, i.e. $\mathcal{S} = \{(x, y, z)|\Psi(x, y, z) = 0\}$. The signed distance transform is used to generate the level set function in our work. This yields an equivalent level set update equation to the surface evolution process in (4.11)

$$\frac{\partial\Psi}{\partial t} = g\left(|\nabla\Phi|\right)\kappa|\nabla\Psi| + \nabla g\left(|\nabla\Phi|\right)\cdot\nabla\Psi. \tag{4.12}$$

Starting with an initial estimate for $\mathcal{S}$ and iteratively updating the level set function using (4.12) leads to a segmentation of the object.

4.4 Results and Applications

In the absence of metric calibration patterns (as is very commonly the case in natural scenes) lifting the requirement for full calibration in every view and relying on homographies induced by a dominant plane (typically the ground) in the scene can greatly simplify the acquisition

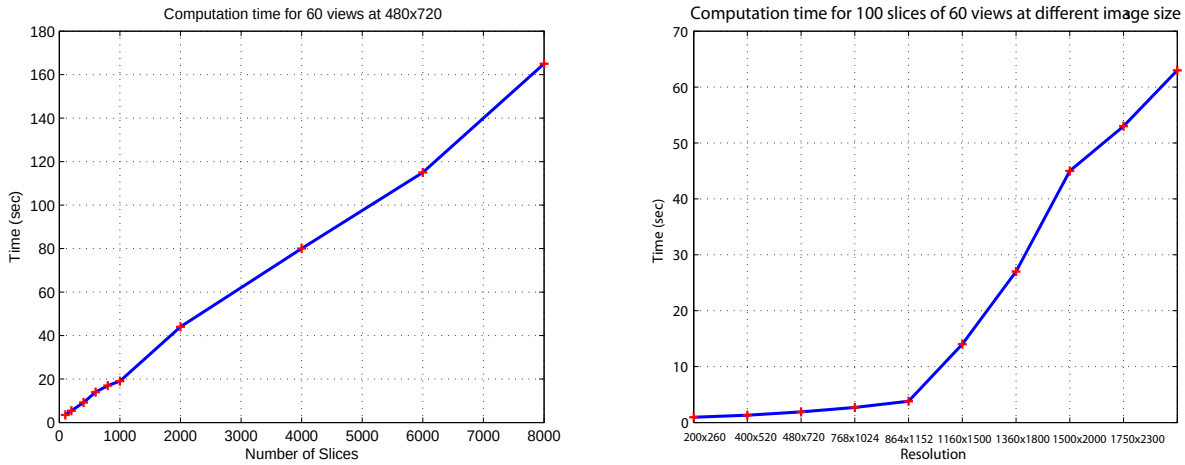


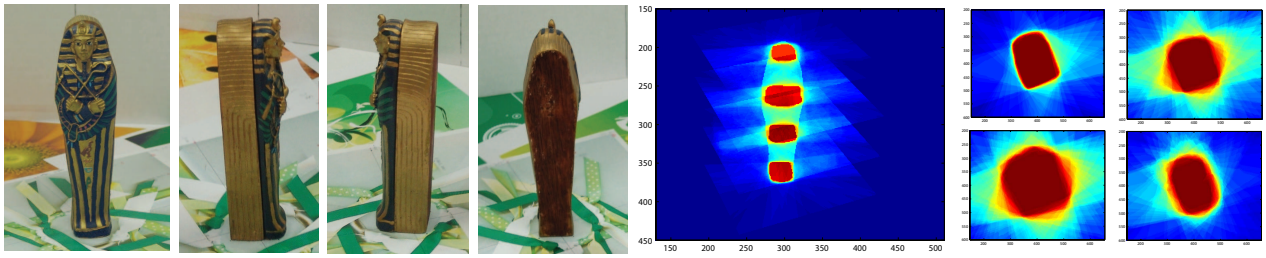
Figure 4.3: Computation time for homographic fusion on a Nvidia Geforce 7300 GPU. (a) Number of slices vs. Time for 60 views each at 480x720. (b) Image Resolution vs. Time for fusing 100 slices from 60 views.

process. A popular scenario is when we have a monocular sequence of a single camera flying around the object in an arbitrary and irregular motion path as is the case in the result shown in figure 4. Also due to the inherent computational simplicity of processing 2D data (conducive to graphics hardware acceleration) our approach has an advantage over other approaches that perform expensive 3D computations, which can be intractable for complex shapes. Our current implementation of homographic fusion runs on a Nvidia Geforce 7300 GPU. It is capable of fusing 60 views (480x720 pixels) at the rate of 50 slices/second (see figure 6).

Though it may be debated that we lose robustness by processing data on cross sections of 3D grids and not on volumetric sections in world space. We believe in cases where full calibration is impractical and computation efficiency important the advantages of our approach convincingly outweigh the reduction in robustness if any. This is corroborated by our experimental results and applications, some of which are discussed next.

4.4.1 Object Reconstruction

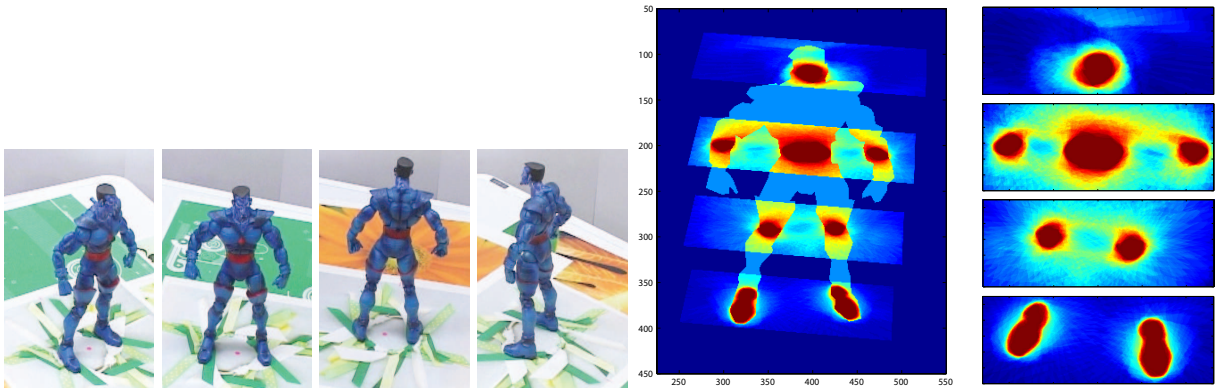
Figures 4.4 and 4.5 show two of the objects that we used in our reconstruction experiments. The data was captured using a digital camera set at a relatively low resolution of 480x720. The mummy sequence in figure 4.4 is a monocular video captured with the camera flying around the object in a very arbitrary/unconstrained motion path (see video sequence in supplementary material). The blue model sequence in figure 5 was captured with the camera stationary and the object on a turntable. Figures 4.4 and 4.5 show 4 of the 30 and 60 views



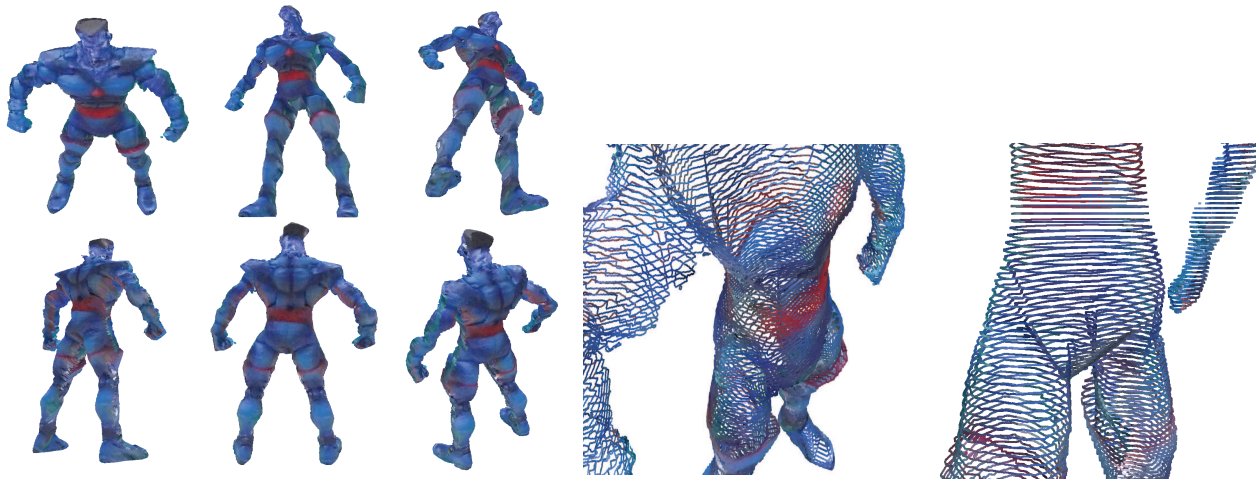
(a) (b)

(c) (d)

Figure 4.4: Structure Modelling: (a) 4 of the 30 views of a mummy statue used in our experiment. (b) The left image is the foreground likelihood map in the reference view with the fusion of 4 of the 200 slices overlaid. Image on the right are the 4 slices shown in log scale (hotter is higher likelihood). (c) Object structure after segmentation from the stacked slices is rendered with point rendering algorithm together with color mapping from the original images. (d) A closeup of segmented slices. The one on the right is showing a view from the bottom of the object looking up.



(a) (b)



(c) (d)

Figure 4.5: Structure Modelling: (a) 4 of the 60 views of an action figure model used in our experiment. (b) The left image is the foreground likelihood map in the reference view with the fusion of 4 of the 200 slices overlaid. Image on the right are the slices shown in log scale (hotter is higher likelihood). (c) Rendering of the object structure after segmentation from the stacked slices. (d) A closeup of segmented slices.

used for each object respectively. In figures 4.4(b) and 4.5(b) we show the reference views in each sequence overlaid with 4 of the 200 occupancy grids/slices computed for each object. The slices are also shown separately in log scale (hotter is higher likelihood). Figures 4.4(c) and 4.5(c) show our reconstruction results. Only contour points of the slice data (after segmentation from Φ) were rendered using a point rendering algorithm. Texture mapping was achieved by reverse warping the slice contour points to the original images for color lookup. Artifacts are visible near the top part of the reconstructions (see head portion of the object in figure 4.5(c)). These are due to small errors in the homographies of the reference plane that get propagated to homographies of the upper planes. In figures 4.4(d) and 4.5(d) we show closeups to emphasize that the reconstruction is slice data rather than a 3D mesh. Notice the detail in which fine curvatures of the objects are captured. It should be pointed out that the result of our method is the affine structure. This is because at no step in the process did we use metric (calibration) information from the scene. Though metric information from the scene can be used to rectify the slice data for full Euclidean structure, it may not be necessary for visualization purposes. For instance, in figure 4.4(c) we used the typical aspect ratio (height to width) of an adult human male. More results from 3D reconstruction algorithm are shown in figure 4.6.

4.4.2 Multiple Object Localization

Our method can be used in much harder conditions for multiple object localization or to initialize a more elaborate photometric method. The presence of high levels of noise, occlusions

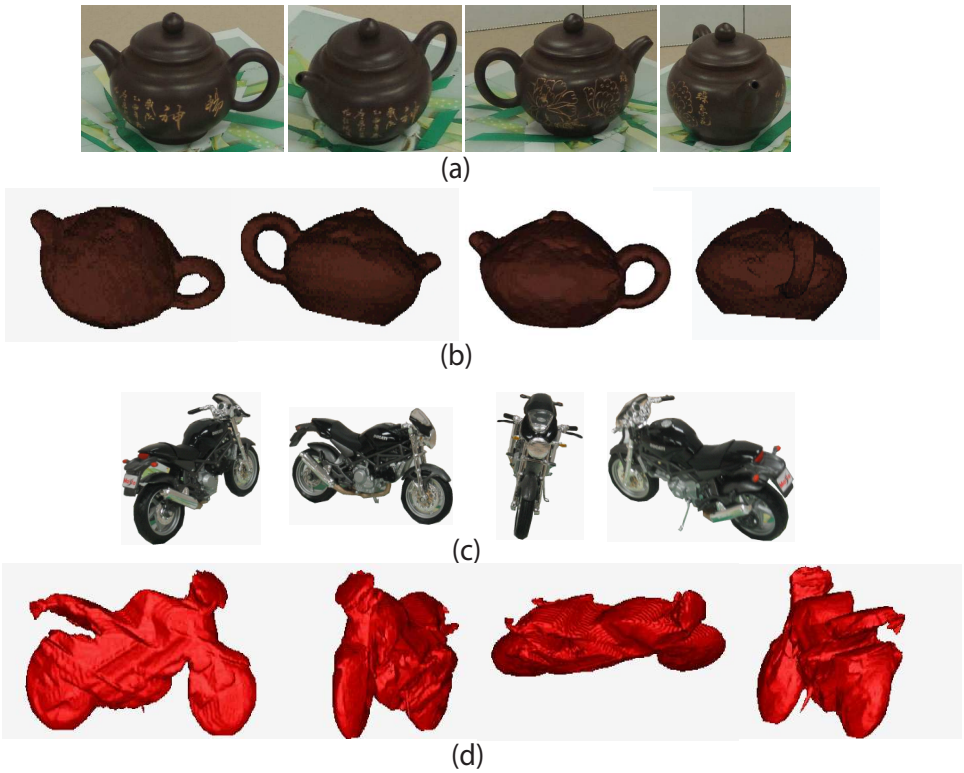


Figure 4.6: Teapot and Motorbike dataset: (a) Four of the 15 views of a teapot. (b) Multiple views of 3D reconstruction of the teapot. (c) Four of the 16 views of a motorbike. (d) Multiple views of 3D reconstruction of the motorbike. A 3D mesh was interpolated on the slice data.

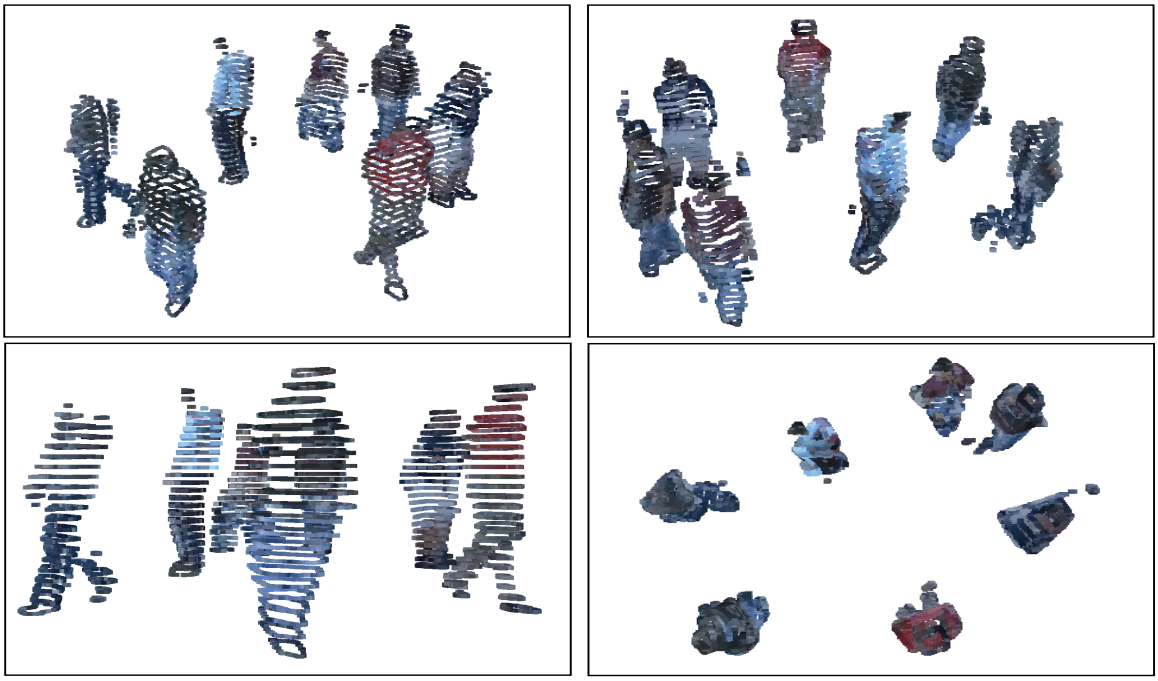
and low resolution limit the use of the method for precise 3D modeling; however, the method can still be used reliably to locate objects in the scene. Our test data is quite challenging as can be seen in figure 4.7(a). It is a surveillance scenario containing multiple people viewed by four wide-baseline cameras covering a relatively large area (parking lot). The cameras have different resolutions and aspect ratios (240x360, 240x320), gamma corrections and the scene has considerably poor contrast, causing noisy background subtraction. The most challenging feature, though, is the severity of inter-occlusions between people limiting the visibility. Due to low resolution on the objects (approx 50 pixels in the longer direction) only a small number of slices could be meaningfully generated. We limited our results to 25 slices. Despite all these factors our method was able to generate surprisingly good results as shown in figure 4.7(b). Though there are a few artifacts (regions not pruned by visual hull intersection), these can be resolved by increasing the number of views.

4.4.3 Multiple Person Tracking

In chapter 3 we presented a tracking approach using the HOC on a single plane. There might be cases where occupancy on the scene reference plane is intermittent e.g. when the ground plane is used and the people are running or jumping resulting in minimal contact with the ground. Or when in the absence of a visible ground plane a building back wall is used where there is in-fact no occupancy, requiring the use of planes parallel to the back wall plane.

As discussed in the pervious section, the HOC can be applied on multiple planes for more robust localization. We therefore use the localization results from the multiple-plane

(a)



(b)

Figure 4.7: (a) The scene contains seven people and is viewed by 4 cameras. Notice the low contrast in the scene that makes background subtraction quite noisy and cluttered. In (b) we show the results of our method. Only 25 slices were computed yet the localization and reconstruction is quite good. Also notice the artifacts in the form of ghost objects. These are due to the lack of visibility created by the inter occlusions and limited number of views. The bottom right image is a top view.

application of the HOC to perform tracking in a 4D spatio-temporal grid. The process is exactly similar to that described in chapter 3, except that we have an additional dimension due to localizations on multiple planes.

4.4.3.1 Video Datasets

We report results on three more multiview tracking datasets. Two of these are novel, challenging datasets that we captured ourselves. The third is a publicly available multi-camera dataset containing video sequences of a soccer match. The frame rate of all the sequences is 30 fps. We computed error rates for these sequences in terms of false detections and missed detections or their sum that we call *detection error*. We define a true positive as one for which, when the bounding box of tracked person (localization of a person over all fusion planes) is transformed from the reference view to other views, it intersects that particular individual in all views. For some of these sequences we calculated detailed tracking errors by comparing with the ground truth (manually marked tracking of the people).

1) Indoor Dataset: This sequence was captured with four frame synchronized cameras (480x720) placed roughly evenly in a semi-circular arc configuration. All four cameras were approximately at head level ∼1.8m (6feet). Due to the camera orientation and configuration the ground plane is only partially visible in just one view. This meant we could not use the ground plane as the reference plane due to a lack of correspondences for homography calculation, a case that might arise in many practical scenarios. We therefore use the back wall in the scene, which is clearly visible in all the views, as the reference plane. Localization

was performed at a total of 20 planes including the wall reference plane and planes parallel to it in the normal direction. Figure 4.8 shows the tracking results for this sequence. The first row shows the bird's eye view with the camera configuration and overlaps in the field of views. As stated for the parking lot sequence, yellower regions have higher camera overlap and the blue squares are tracked locations.

Figure 4.9 shows the quantitative analysis for this sequence. We did not have metric calibration data for this sequences therefore we calculated the total average track error in the image space. This was done by calculating the distance in pixels between the top of the tracked localization (centroid of top patch of the track bounding cubes in figure 4.8) and manually marked the top of the heads of the people.

Figure 4.9(a) shows the variation of the total average track error for a sequence of frames by selectively varying the number of views. With four views the track error hovers around 20 pixels, which is quite good considering the size of a person is about 250x75 pixels, and only the head location rather than a central axis was used to calculate the track error. With only two views though the tracking becomes intractable. The magenta plot in 4.9(a) also shows the track error for four views if clutter modelling is not used. As can be seen the total average track error can be reduced by nearly 10 pixels if clutter is modelled. In figure 4.9(b) we show the plots of *accumulated* detection errors (sum of false positives and false negatives accumulated over time) if only a single fusion plane is used for the HOC. Results of 4 of the 20 planes including the back wall plane and planes parallel to it are shown. The original reference plane i.e. the back wall has the worst performance (red plot) because at no time

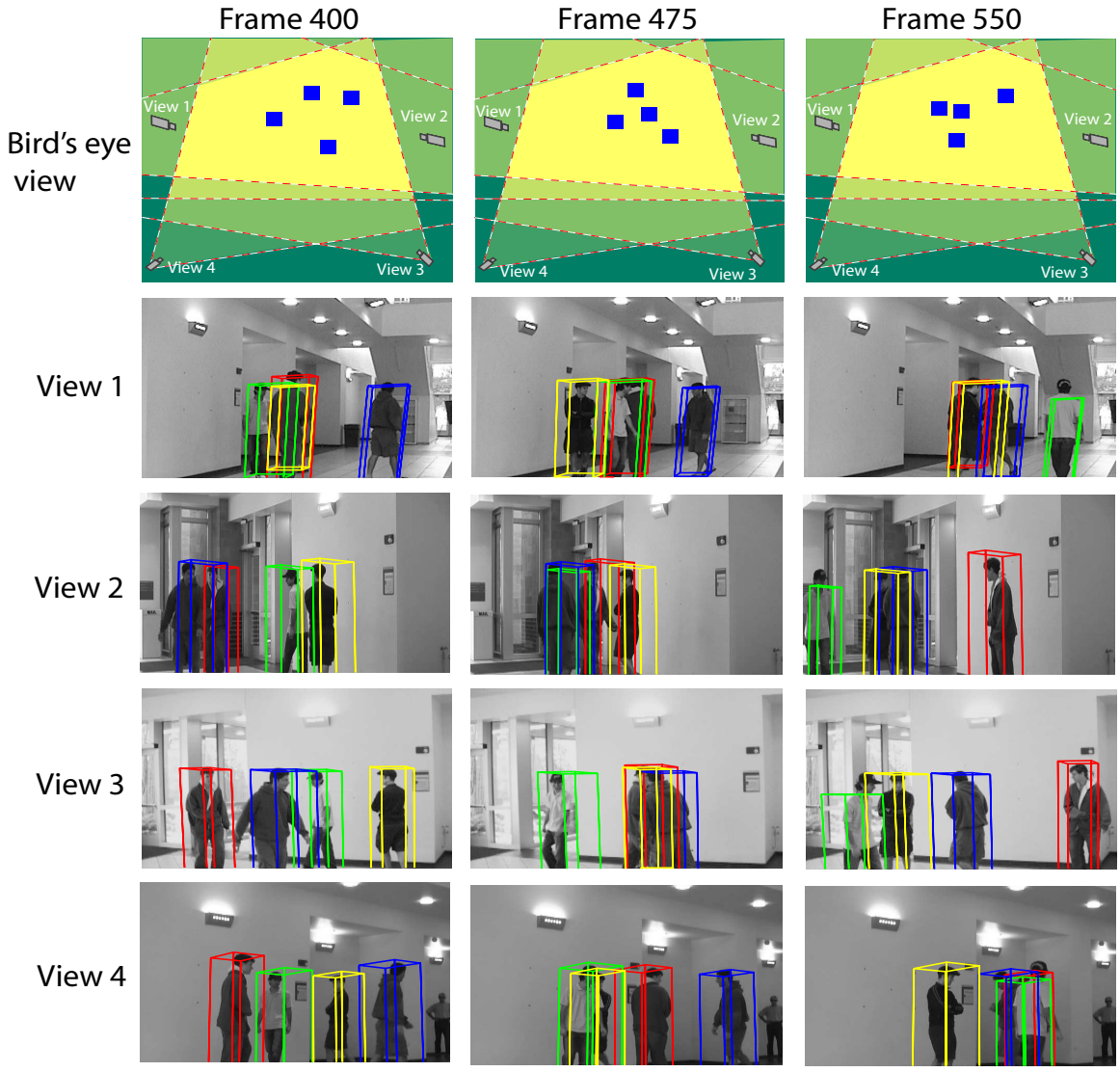


Figure 4.8: *Indoor DataSet:* Tracking using the back wall as the primary reference plane. 20 planes parallel to the back wall were used in total. The top view color coding is the same as in figure 3.7. 3D bounding boxes encapsulating the localization on all fusion planes are plot.

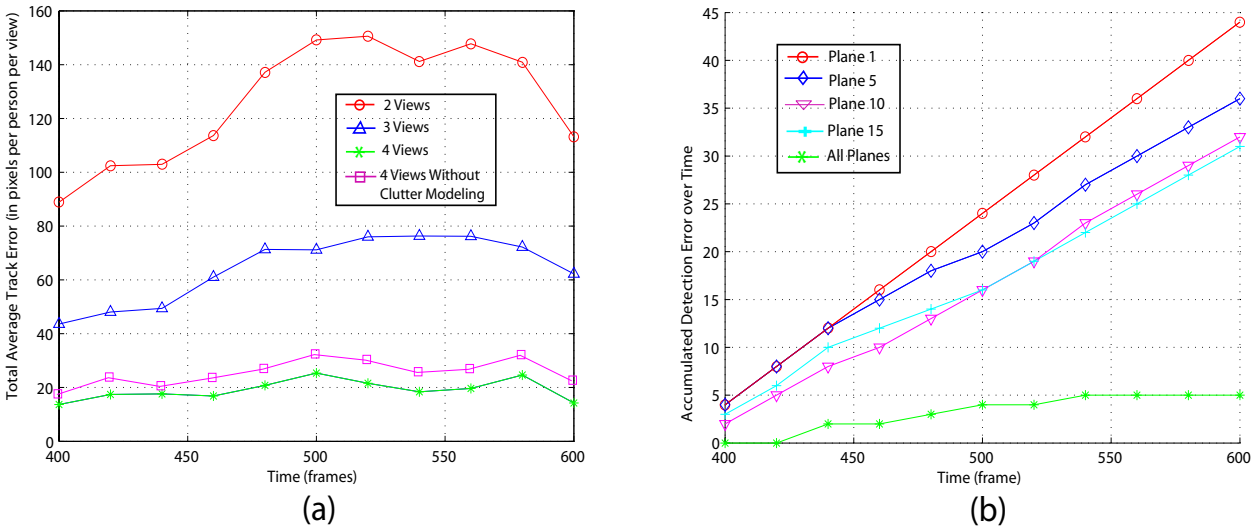


Figure 4.9: *Indoor Dataset Analysis:* (a)Total average track error over time, for the top center of the track bounding box from manually marked head locations of people. (b) Plot on the top shows the accumulated detection error (number of false positives + number of false negatives accumulated over time) for different individual planes. Error is the worst for plane 1 i.e. the back wall since at no time are people touching (occupying) the back wall. Other planes parallel to the back wall in the normal direction are only marginally better. This is because people keep moving away and towards the back wall in circles, meaning there is no one single plane that can be used to reliably localize the people. Since we use all the plane simultaneously our localization errors are significantly reduced as shown by the green plot.

instant were there people touching (occupying) the back wall, resulting in zero detections. Other individual planes fare only marginally better. The people kept moving in circles, coming closer and going farther from the back wall. This meant there was no single plane that could reliably localize all people over a meaningful period of time. Figure 4.9(b) also shows in green the plot of accumulated localization errors when using all 20 fusion planes together. As can be seen the error is significantly reduced, thus corroborating our initial motivation to use multiple planes to localize people.

2) Basket Ball Dataset: This dataset was captured with 3 cameras (480x720), arranged roughly in a semi-circular arc. The sequences are approximately 1000 frames long and consists of 10 players playing basketball. HOC fusion was applied on 20 planes including the ground plane and planes parallel to it in the up direction. Based on observation the highest plane was approximately 8feet (2.5m) above the ground.

This dataset is challenging not only because of the limited number of views and occlusions due to the high number of players, but also because of shadows and reflections off the shinny floor. Moreover the players have highly non-linear and unpredictable motion paths including jumps and leaps off the ground. For other approaches, even if occlusions are resolved and the shadows and reflections removed, it will still be difficult to keep track of the players with such motion paths. Whereas our approach performs quite well, see figure 4.10. Notice especially the player who is jumping and being tracked (red bounding box, second column of fig 4.10). Clearly using a single fusion plane like the ground, would cause the

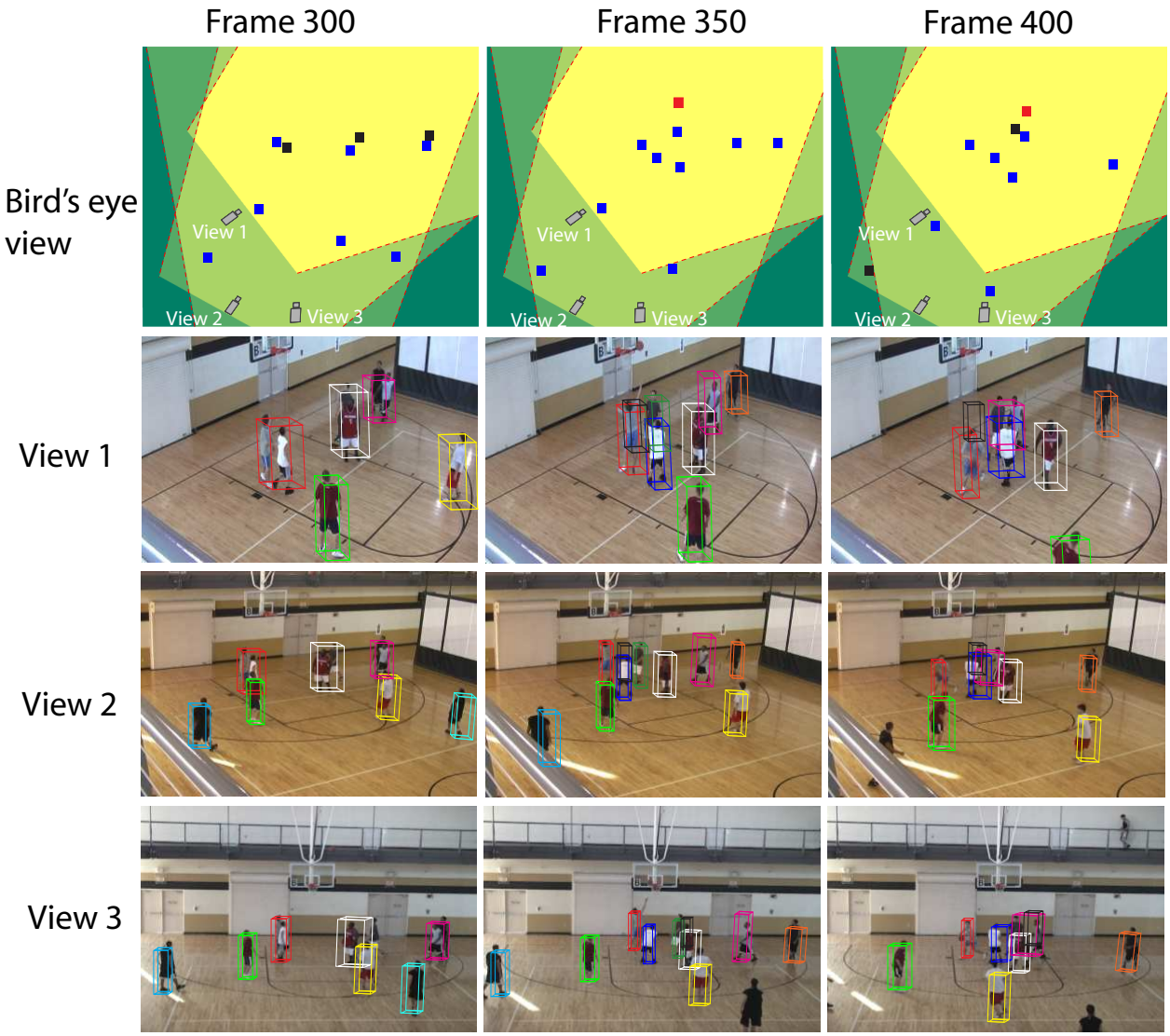


Figure 4.10: *Basket Ball Dataset:* Tracking of multiple players in a basket ball game. Notice the track of player who is jumping (red track box in frame 350). Due to limitations in the number of cameras and constraints on camera configuration as well as scene clutter due to reflections off ground and occlusions, our results had relatively higher detection errors. See red, white and magenta track boxes in views corresponding to frame 300. One player is missed in each box (black squares in top view). Also there are some false positives in frames 350 and 400 (red squares in top view, and black track boxes in camera views).

jumping player's track to be lost. Though there are false positives and false negatives (red and black squares in the top view), we believe more views can resolve many of these. The false negative rate for the sequence (calculated using ground truth at intervals of 50 frames over 500 frames) is $\sim 17\%$, and the false positive rate is $\sim 8\%$.

3) Soccer Dataset: This a publicly available dataset [114], consisting of 8 frame-synchronized views of a soccer match. The cameras are placed in a configuration that covers the entire pitch with two focused on the goal areas on opposite sides. The sequences are about 1000 frames long. Occlusions are quite abundant due to the large number of players. There is also a lot of clutter due to jitter in the cameras. We believe this is a result of winds or the shaking of the platform on which the cameras were mounted. Another challenge is the lack of pixel resolution on players. Depending on the view, player patches could be as small as 5x25 pixels. In spite of these challenges our method was able to localize and track the players with a high degree of accuracy. Figure 4.11 shows our tracking results. The first row is the top view with the same color coding as previously described for other sequences. Notice there are greater instances of errors in regions of less view overlap and higher density of players.

4.5 Summary

In this chapter we have presented an image-based visual hull approach for fusing foreground silhouette information from multiple views to generate 3D structure of objects. Unlike other

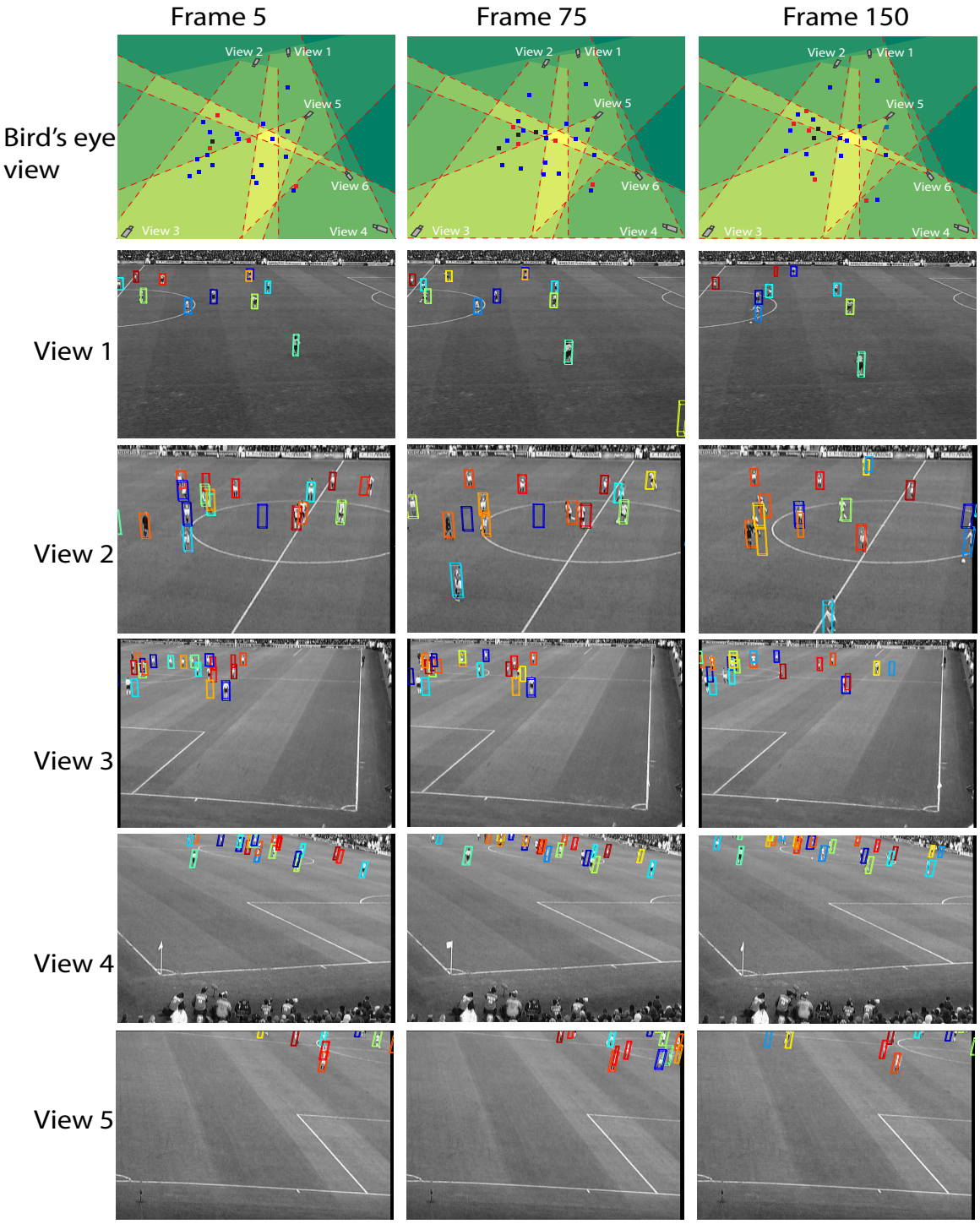


Figure 4.11: *Soccer Dataset:* Tracking of multiple players in a soccer match. The top view is color coded as in earlier figures. In rows 2-6 we show views 1-5 of the available views. Due to limitations on space and adequate visualization, all views could not be shown.

visual hull based methods that require calibrated views and use 3D constructs like voxels, 3D visual cones or polygonal meshes, our method utilizes the HOC requiring only minimal geometric information i.e. homographies between views and the vanishing points of a reference direction. The HOC is shown to implicitly perform visual hull intersection in the image plane without without requiring to go in 3D space. This process is extended to multiple planes parallel to the reference plane, each computation delivering a successive cross-sectional slice of the object. We have also used to this approach to increase the robustness and applicability of our tracking algorithm. We are now able to perform tracking in previously unexplored scenarios like when contact with the reference plane is intermittent (people jumping or running) or the ground plane is not visible and a different plane like a building wall needs to be used.

One limitation of this approach, and for that matter any visual hull based approach is that in monocular video non-stationary objects cannot be reconstructed. In such a scenario, silhouettes of the object no longer constrain the visual hull of the object and moving parts of the object are in advertently carved out by visual hull intersection. To address this problem, in the next chapter, we present a new approach to 3D reconstruction on non-stationary articulated objects in monocular video.

CHAPTER 5
3D RECONSTRUCTION ON NON-STATIONARY ARTICULATED OBJECTS IN MONOCULAR VIDEO

5.1 Introduction

In this chapter we present a novel approach to reconstruct the 3D shape of an object from silhouettes obtained in a monocular video sequence with the object undergoing rigid or non-rigid motion [4]. The homographic occupancy constraint (HOC) as developed and utilized in the conventional sense can be used for reconstruction and localization in a monocular video sequence if the object is stationary (rigid). Traditionally visual hull based approaches rely on object silhouettes obtained from multiple time-synchronized cameras or if a single camera is used for a fly-by (or a turn table setup) the scene is assumed to be static. These constraints greatly limit the applicability of visual hull based approaches to controlled laboratory conditions. In real-life applications, a sophisticated multiple-camera setup may not be available. If a single camera is used to capture multiple views by going around the object, it is not reasonable to assume that the object will remain static over the course of time it takes to obtain views of the object, especially if it is a person, animal or vehicle on the move.

Though in the past there has been some work on using visual hull reconstruction in monocular video sequences of rigidly moving objects to recover shape and motion [126][122][116],

these methods involve the estimation of 6 DOF rigid motion of the object between successive frames. To handle non-rigid motion the use of multiple cameras becomes indispensable [117]. Unlike these approaches we do not require the detection of surface feature points in 3D (frontier points, colored surface points) for the estimation and eventual compensation of the motion of the scene object. Rather we introduce the concept of motion blurred scene occupancies, a direct analogy of the motion blurred image but in a 3D object scene occupancy space. Similar to a motion blurred picture caused by the movement of a scene object (or the camera) and the camera sensor accumulating scene information over the exposure time, 3D scene occupancies will be mixed with non-occupancies where there is motion resulting in a *motion blurred occupancy space*. By de-blurring this data with appropriate point spread functions (PSF), we are able to obtain the motion compensated 3D shape of the object. Note that our approach is different from the traditional structure from defocus/deblur approaches [120] [123]. There the objective is to obtain a depth map/surface of the scene from one or more blurred (out of focus) appearance images by adjusting camera focal length, or recovering motion of the object that caused the motion blurred appearance.

Our approach takes a different route to recover structure from multiple views obtained from a monocular video sequence of a non-stationary object. Instead of using motion blurred appearance image/s we fuse silhouette information from multiple views to create motion blurred scene occupancy information, where greater blur (lesser occupancy value) is interpreted as greater mixing of occupancy with non-occupancy in the total time duration. We

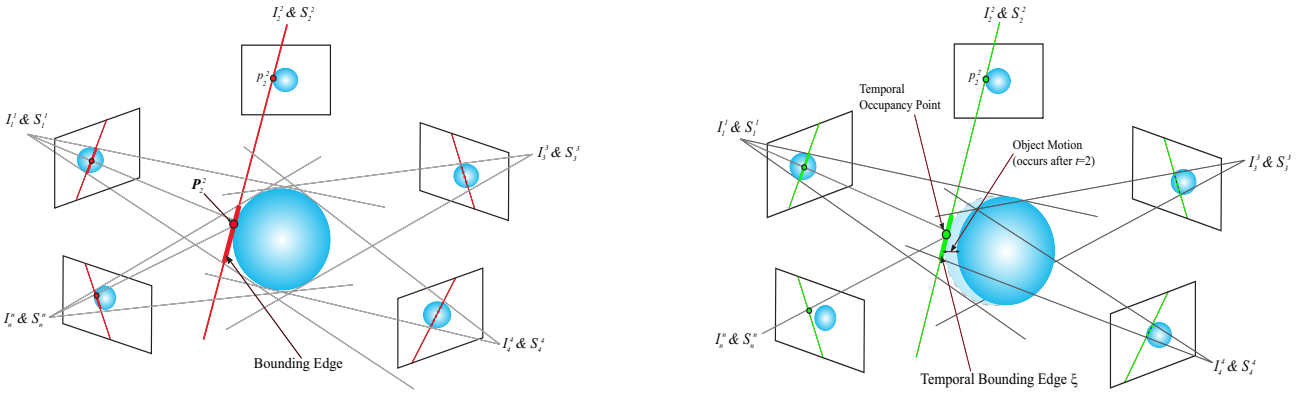
then use a motion deblurring approach to obtain the mean/motion compensated 3D shape of the scene object over the duration of time.

5.2 Approach

Silhouette information has been used in the past to estimate occupancy grids for the purpose of detection and reconstruction. Due to the inherent nature of visual hull based approaches if the silhouettes correspond to a non-stationary object obtained at different time steps (monocular video), grid locations that are not occupied consistently will be carved out. As a result the reconstructed object will only have an internal body core (consistently occupied scene locations) survive the visual hull intersection. Our first task is therefore to identify occupancy grid locations that are occupied by the scene object and for the durations that they are occupied. In essence scene locations giving rise to the silhouettes in each view need to be estimated.

5.2.1 Obtaining Scene Occupancies

Let $\{ I_t^n, S_t^n; n = 1, ..., N\}$ be the set of color and corresponding foreground silhouette information generated by a stationary object $\mathbf{O}$ in N views obtained at times $t = 1..., T$ in a monocular video sequence (e.g. the camera flying around the object). Let p_j^i be a pixel in the foreground silhouette image S_j^i. With the camera center of view i, p_j^i defines a ray r_j^i in 3D space. If the scene object is stationary, then a portion of r_j^i is guaranteed to project inside the bounds of the silhouettes in all the image planes, and in past literature it has



(a) (b)

Figure 5.1: In case of a stationary object we can obtain the bounding edge for a pixel on the foreground silhouette by extending a ray through the pixel and selecting the section of the ray that projects to within the bounds of silhouettes in all views. This process is shown in (a) where the bounding edge corresponding to pixel p_2^2 in view I_2^2 is highlighted with a bold red segment of the red ray. When the object is undergoing motion the ray through a silhouette pixel is not guaranteed to project to within the bounds of silhouettes of other views. In this case for pixel p_2^2 we have a temporal bounding edge which is the section of the ray through p_2^2 that projects to the highest number of silhouettes as shown in (b). The temporal occupancy point corresponding to p_2^2 is also shown. This is the point on the temporal bounding edge that when projected in the visible images has minimum color variance and is good estimate of the 3D scene point that is imaged at p_2^2.

been referred to as the *bounding edge* [116], see figure 5.1(a). Assuming the object to be Lambertian and the views to be color balanced, the 3D scene point P_j^i corresponding to p_j^i can be estimated by searching along the bounding edge for the point with minimum color variance when projected to the *visible* color images.

Now, if object **O** is non-stationary and P_j^i is not consistently occupied over the time period $t = 1 : T$ then r_j^i is no longer guaranteed to have a bounding edge. There may be no point on r_j^i that projects to within object silhouette in every view, in fact there may be views where r_j^i projects completely outside the bounds of the silhouettes as shown in figure 5.1(b). Since the images are obtained sequentially at different time instances, the number of views in which r_j^i projects to within the bounds of silhouettes would in turn put an upper bound on the amount of time (w.r.t. to total duration of video) P_j^i is guaranteed to be occupied by **O**. Let us define as *temporal occupancy* τ_i^j, the fraction of total time instances T (views) where r_j^i projects to within the bounds of a silhouette and temporal bounding edge ξ_j^i as the section of r_j^i that this corresponds to as shown in figure 5.1(b). Now we can formally state aforementioned ideas in the following proposition:

Proposition For a silhouette point p_i that is the image of scene point P_i, τ_i provides an upper bound on the duration of time it is guaranteed to be occupied and determines the temporal bounding edge ξ_i on which P_i must lie.

In the availability of scene calibration information, ξ_j^i and τ_j^i can be obtained by succes-

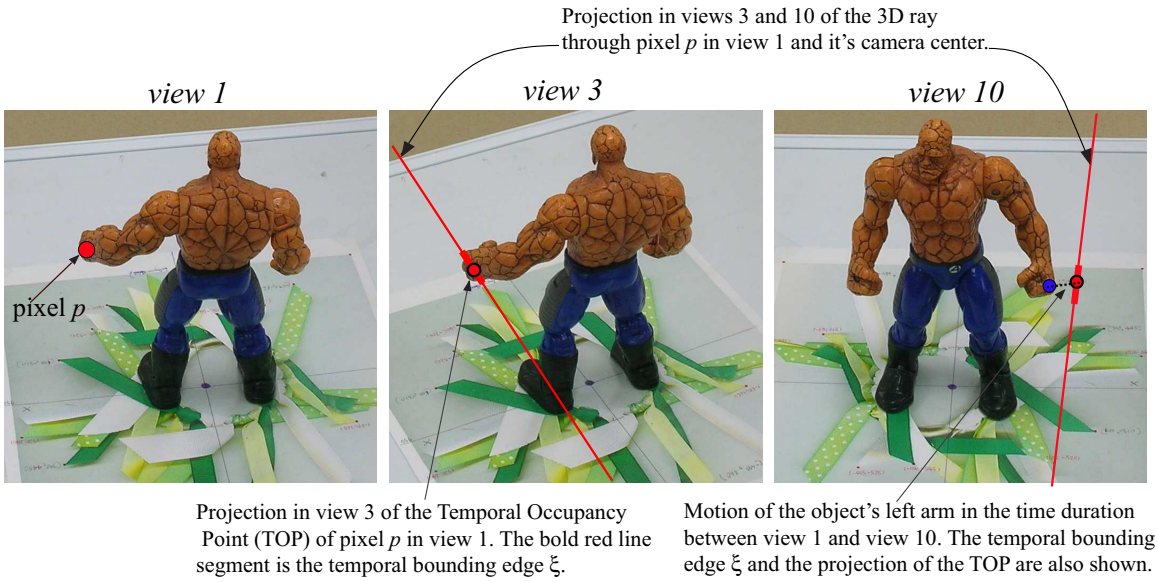


Figure 5.2: Three frames from a monocular sequence of a non-rigidly deforming object (motion in the left arm after view 3). The 3D ray corresponding to the pixel marked with a red circle in view 1 is projected in views 3 and 10. Notice that due to the motion of the object the projected ray does not pass through the object silhouette in view 10. The projection of the pixel's temporal bounding edges and TOPs are also shown in views 3 and 10.

sively projecting r_j^i in the image planes and retaining the section that projects to within the maximum number of silhouette images. To refine our localization of the 3D scene point P_j^i (corresponding to the silhouette pixel p_j^i) along ξ_j^i, we develop another construct called the *temporal occupancy point* obtained by enforcing the appearance/color constancy constraint as described in the next section.

5.2.1.1 Temporal Occupancy Points

If it can be guaranteed views of the object are captured faster than it's motion, then without loss of generality the non-stationary object $\mathbf{O}$ can be considered piece-wise stationary: $\mathbf{O} = \{ \mathbf{O}_{1:s_1}, \mathbf{O}_{s_1+1:s_2},..., \mathbf{O}_{s_k:T}\}$, where each s_i marks a time where there is motion in the object. This assumption is easily satisfied in high capture rate videos where for small batches of frames non-stationary objects tend to be rigid. With the previous assumptions of Lambertian surfaces and color balanced views, having piece-wise stationarity would justify a photo-consistency check along the temporal bounding edge for scene point localization. We can then proceed with a linear search along the temporal bounding edge ξ_j^i for a point that touched the surface of the object. Such a point will have the property that it's projection in the *visible* images has minimum color variance (color constancy constraint). We refer to this point as the *Temporal Occupancy Point (TOP)* as shown in figure 5.1(b), and use it as the estimated localization of the 3D scene point P_j^i that gave rise to the silhouette pixel p_j^i.

In figure 5.2 we demonstrate this process on some real data used in our experiments. The figure shows three views from a monocular camera sequence (flyby) as the object moves it's left arm. Pixel p marked with a red circle corresponding to the left hand in view 1 is selected for demonstration. The 3D ray back projected through this pixel is imaged in views 3 and 10, shown by the red lines. Notice that due to the motion of the object (left arm moving down) in the time duration between views 1 and 10, the ray does not pass through the corresponding left hand pixel in view 10 (marked with blue circle). In fact the projection of the ray is completely outside the bounds of the object silhouette in view 10. The temporal

bounding edges and the TOPs corresponding to pixel p are computed and their projections in view 3 and 10 are also shown.

Since we are using monocular video sequences, it may not be the case that we have complete camera calibration at each time instant, particularly if the camera motion is arbitrary. Our strategy is therefore to use a purely image-based approach. For each silhouette pixel, instead of determining it's corresponding TOP explicitly in 3D space, we directly obtain the projections (images) of the TOP in each view. If the object was stationary and the scene point visible in every view, then a simple stereo based search algorithm could be used. Given the fundamental matrices between views, the ray through a pixel in one view can be directly imaged in other views using the epipolar constraint [132]. The images of the TOP can then be obtained by searching along the epipolar lines (in the object silhouette regions) for a correspondence across views that has minimum color variance. Since neither the object is stationary nor the scene point guaranteed to be visible from every view, the stereo based approach described above is not viable. As can be seen in figure 5.2, pixel p in view 1 has an epipolar line in view 10 (image of the projected ray through p) that is outside the bounds of the object silhouette. Therefore, it is not possible to search along the epipolar line for a correct correspondence. The same can be argued for cases when the scene point is not visible from a view (self occlusion of the object). We, therefore, propose to use homographies induced between views by a pencil of planes for a point to point transformation.

Consider figure 5.3. The image of the 3D scene point P_ϕ (corresponding to the image point p_{ref} in the reference view) can be directly obtained in other views by warping p_{ref}

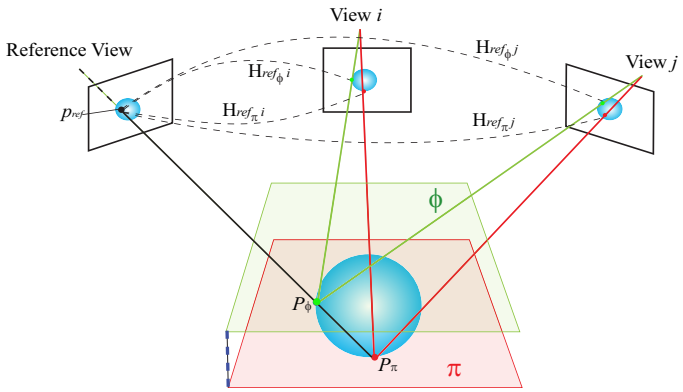


Figure 5.3: In the absence of complete camera calibration, 3D scene points on a ray passing through a pixel can be directly imaged in other views by warping the pixel with homographies induced between views by a set of parallel planes intersecting the ray. If the ground plane is used as the reference plane, homographies of successively parallel planes can be obtained using the vanishing point of the normal/up direction.

with the homography induced by a plane ϕ that passes through P_ϕ as shown in figure 5.3. To obtain this homography, we can use a ground plane reference system. Given the homography induced by a ground scene plane and the vanishing point of the normal direction, homographies of planes parallel to the ground plane in the normal direction can be obtained using the relationship developed in equation 4.8.

The projection of the temporal bounding edge ξ_j^i in the image planes can be obtained by warping p_j^i with homographies of successively higher planes (by incrementing the value of γ) and selecting the range of γ for which p_j^i warps to within the largest number of silhouette images. The image of p_j^i's TOP in all the other views is then obtained by finding the value of γ in the previously determined range, for which p_j^i and its homographically warped locations have minimum color variance in the visible images. The upper bound on occupancy duration τ_j^i is evaluated as the ratio of the number of views where ξ_j^i projects to within silhouette boundaries and the total number of views. This value is stored at the imaged locations of the TOP.

Our visibility model specifies which views to consider when evaluating photo-consistency measures and is a combination of geometric and quasi-geometric approaches. We start with a popular heuristic of limiting the photo-consistency analysis to clusters of nearby cameras/views [137][138]. This has two advantages, firstly by choosing nearby views we are minimizing the effects of occlusions as view points close to each other will have similar scene visibility. Secondly, in monocular video of non-stationary objects, by limiting ourselves to clusters of nearby views we are approaching piece-wise stationary, thereby making the photo-

consistency check more reliable and robust. The cluster of views is used to perform visual hull intersection in the image plane in the manner described in chapter 4. This delivers a slice of the object in the view where visibility needs to be checked. Now to check the visibility of a point in the view, we simply translate the point in the direction of the up vanishing point (normal to the ground reference plane). If the point moves into the disk of the slice we know that the point is not visible from the view. But if the point moves outside the disk of the slice, it means the scene point is facing the camera and is visible.

5.2.1.2 Building Blurred Occupancy Images

As described above, for a silhouette pixel we can obtain the image location of its TOP in every other view. We uniformly sample the boundary of the object silhouette in each view and project their TOPs in all the views. The accumulation of these projected TOPs delivers a corresponding set of images that we call the blurred occupancy images: $B_t^n; n = 1, ..., N; t = t, ..., T$. The pixel values in each image are the occupancy durations τ of the TOPs. Examples are shown in figure 5.4 and the analogy with motion blurred images is apparent. Due to the motion of the object, regions in space are not consistently occupied resulting in some occupancies blurred out with non-occupancies which is reflected in the blurred occupancy images. The algorithmic procedure is described in the following steps:

Objective Generate blurred occupancy images B_t^n.

- **for** each silhouette image

- Uniformly sample silhouette boundary
- **for** each sampled silhouette pixel p
  - Obtain temporal bounding edge ξ and occupancy duration τ
    - As described in 5.2.1.1 transform p to other views using multiple plane homographies.
    - Select range of γ (planes) for which p warps to within the silhouette boundaries of the largest number of views.
  - Find projected location of TOP in all other views
    - Obtain cluster of visible views
    - Search along ξ (values of plane γ)
    - Project point to visible views
    - Return if minimum variance in appearance amongst the views.
  - Store value of τ at projected locations of TOP in the Blurred occupancy images B_t^n.
- End **for**.

- End **for**.

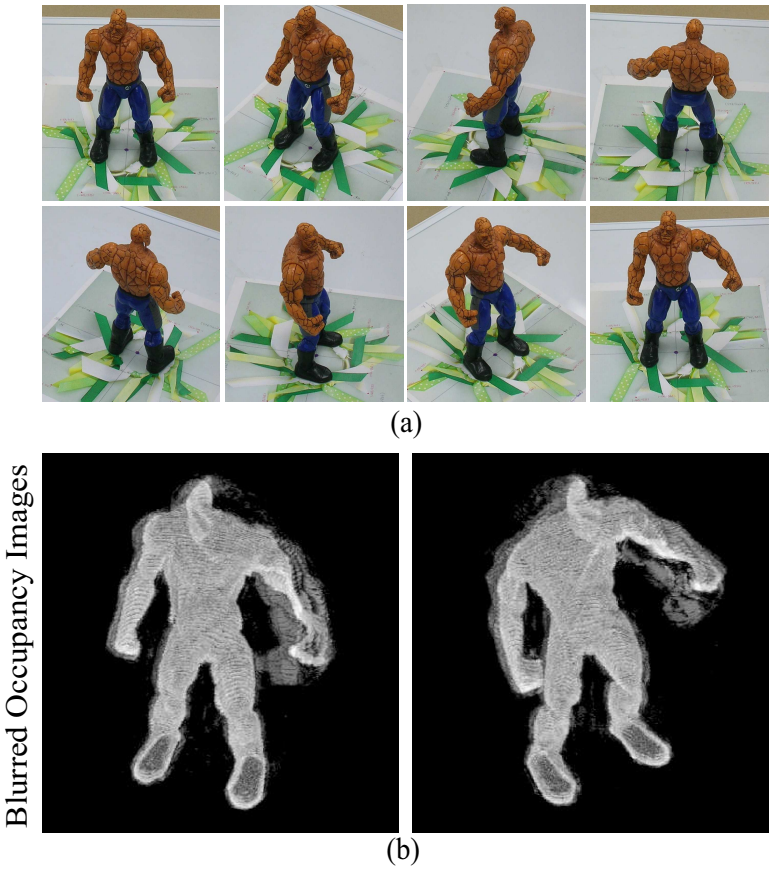


Figure 5.4: (a) Eight of the 20 views used in this dataset. Notice the left arm of the model is moving (compare first and last images). (b) Two of the *blurred occupancy images*. Due to the motion of the arm some sections of the scene (where the moving arm passes through) are not consistently occupied. This results in scene occupancies mixed with non-occupancies which generates the blurred silhouette/occupancy image. The section of the arm that has the greatest motion is also shown in cropped and zoomed views in the second row.

5.2.2 Motion Deblurring

The motion blur in the blurred occupancy images can be modelled as the convolution of a blur kernel with the latent occupancy image plus noise:

$$B = L \otimes K + n, \tag{5.1}$$

where B is the blurred occupancy image, L is the latent or unblurred occupancy image, K is the blur kernel also known as the point spread function (PSF) and n is additive noise. Conventional blind deconvolution approaches focus on the estimate of K to deconvolve B using image intensities or gradients. In traditional images, there is the additional complexity that may be induced by the background that may not undergo the same motion as the object. The PSF has a uniform definition only on the moving object. This however is not a factor in our case since the information in blurred occupancy images corresponds only to the motion of the object. Here we do not propose a new method to compute the blur PSF since that is not the focus of this work and there have been several successful blind deconvolution algorithms developed in the recent past [134][133][135]. We use an approach similar to the recent work by Jia [136]. They first segment the foreground object as a blurred transparency layer and use the transparency information in a MAP framework to obtain the blur kernel. By avoiding taking all pixel colors and complex image structures into computation their approach has the advantage of simplicity and robustness but requires the estimation of the object transparency or alpha matte. The object occupancy information in our blurred occupancy maps once normalized in the [0-1] range and can be directly interpreted as the transparency information or an alpha matte of the foreground object.

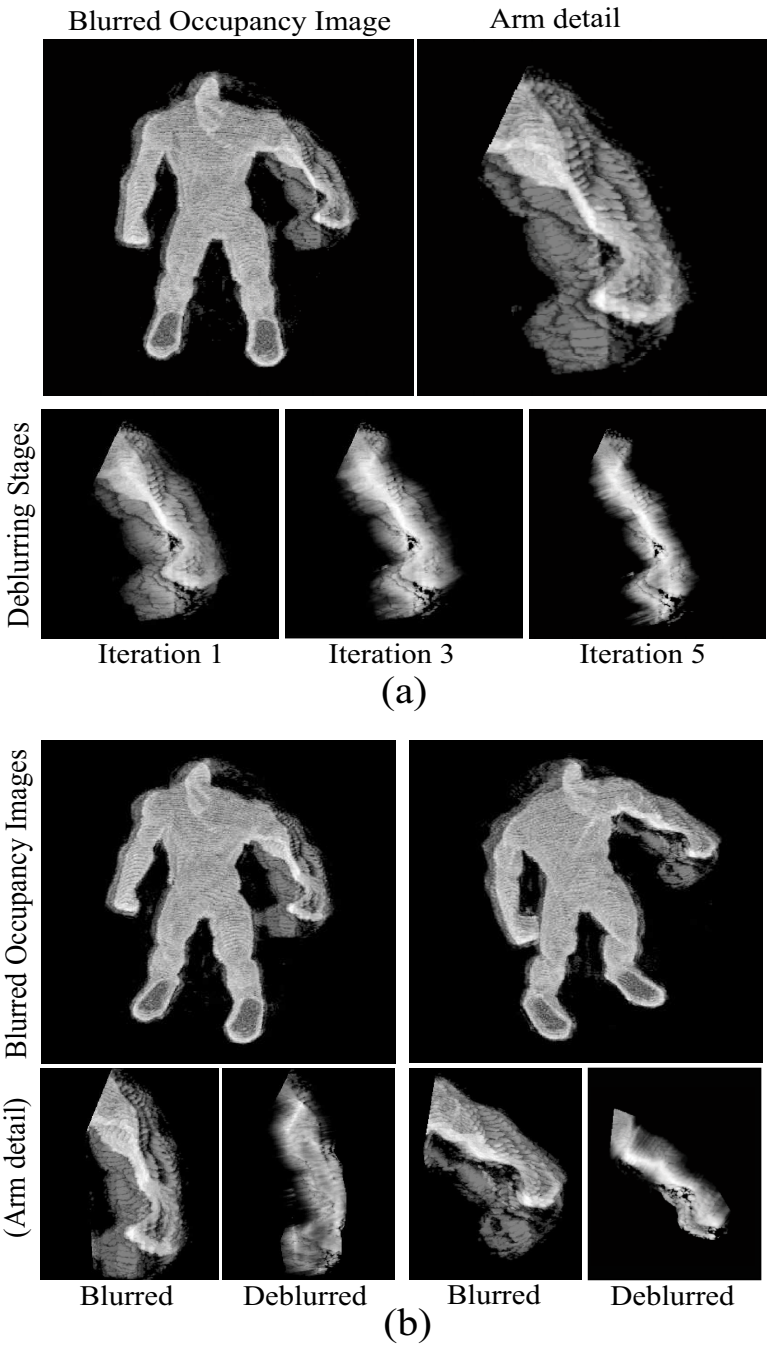


Figure 5.5: (a) Top left is a blurred occupancy image generated from one of our experiments and top right is the cropped section on the arm. In the second row we show the deblurring results after 1, 3 and 5 iterations of the deconvolution process. The initial estimate of the blur kernel was a horizontal motion filter. (b) More examples of blurred occupancy images and the final deblurred results.

The blur filter estimation maximizes the likelihood that the resulting image, when convolved with the resulting PSF, is an instance of the blurred image, assuming Poisson noise statistics. The process de-blurs the image and refines the PSF simultaneously, using an iterative process similar to the accelerated, damped Lucy-Richardson algorithm. We start with an initial guess of the PSF as simple translational motion. This is fed into the blind deconvolution approach that iteratively restores the blurred image and refines the PSF. In figure 5.5(a) we show the deblurring in various stages of the process. Notice the confluence of the blurred occupancies into a single cohesive deblurred region with increasing iterations.

It should be noted that our deblurring approach assumes uniform motion blur but that may not be the case in natural scenes. For instance due to the difference in motion between the arms and the legs or a walking person the blur patterns in occupancies may be different and hence different blur kernels will need to be estimate for each section. This is a very challenging problem and though there is some very recent work on estimating blur kernels in the case of non-uniform blurring [139], this problem is beyond the scope of this work and will be addressed in future studies. In our method for PSF estimation the user specifies different crop regions of the blurred occupancy images each with uniform motion, which are then restored separately. After estimating the blur filters, we apply the Lucy-Richardson method to deconvolve the blurred occupancy images to obtain the de-blurred occupancy maps $L_t^n; n = 1, ..., N; t = 1, ..., T$, which are used in the final reconstruction.

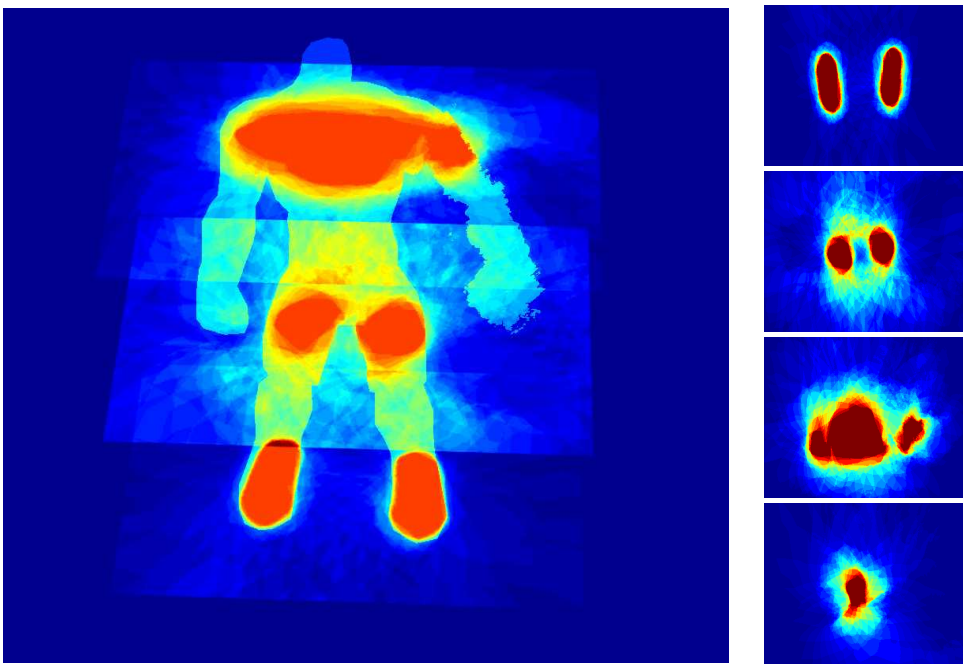


Figure 5.6: After deblurring these are used to perform a slice based reconstruction of the object (b) Three of the 100 slices are overlaid onto a reference view (deblurred occupancy map). (c) The slices are shown separately.

5.2.3 Final Reconstruction

Once motion deblurred occupancy images have been generated, the final step is to perform a probabilistic visual hull intersection. Conventional approaches can be used [130] and for our purposes the homographic framework for 3D reconstruction presented in chapter 4 [121] is ideal as it handles arbitrary camera motion without requiring full calibration.

As described in chapter 4 the 3D structure of objects is modelled as being composed of an infinite number of cross-sectional slices, with the frequency of slice sampling being

a variable determining the granularity of the reconstruction. Using planar homographies induced between views by a reference plane in the scene (ground) occupancy maps L_is' (foreground silhouette information) from all the available views are fused into a reference view (arbitrarily chosen) performing visual hull intersection in the image plane. This process delivers a 2D grid of object occupancy likelihoods representing a cross-sectional slice of the object. Consider a reference plane π in the scene inducing homographies $H_{i_\pi j}$, from view i to view j. Warping L_is to a occupancy map in a reference view L_{ref} , we have the warped occupancy maps: $\hat{I}_i = [H_{i_\pi j} L_i]$. Visual hull intersection on π is achieved by fusing the warped occupancy maps:

$$\phi_{ref} = \prod_{i=1}^{n} \hat{L}_i, \tag{5.2}$$

where ϕ_{ref} is the projectively transformed grid of object occupancy likelihoods or an object slice. Notice how using this homographic framework visual hull is being performed in the image plane without requiring to go in 3D space.

Subsequent slices or ϕs of the object are obtained by extending the process to planes parallel to the reference plane in the normal direction. Homographies of these new planes can be obtained using the relationship in equation 4.8. Occupancy grids/slices are stacked on top of each other creating a three dimensional data structure: $\Phi = [\phi_1; \phi_2; \ldots \phi n]$ that encapsulates the object shape. Object structure is then segmented out from Φ i.e., simultaneously from all the slices by evolving a smooth surface $\mathcal{S} : [0, 1] \rightarrow \mathbb{R}^3$ using level sets that divides Φ between the object and the background (see chapter 4 for details).

5.3 Results and Experiments

We have tested our approach on several challenging monocular datasets. Figure 5.4 shows our 'The Thing' dataset. It consists of 20 views of a humanoid model captured with a camera moving around the object while the object deforms non-rigidly. (The left arm of the model is moving). In figure 5.4(b) two of the twenty blurred occupancy images (one in each view) produced using our approach are shown. Notice the occupancies in the region surrounding the left arm are blurred due to the motion of the arm. This region is selected and shown with more detail in the images in the second row of 5.4(b) together with the deblurred results. The motion deblurred occupancy images are then used to reconstruct the object using the image-based approach as described in section 3.3. A hundred slices were used to reconstruct the object. In figure 5.6(a) the reference view (from the deblurred occupancy images) used in reconstruction is shown with three of the hundred slices overlaid. The slices are also shown separately in log scale in 5.6(b) (redder is higher likelihood).

Figure 5.7(a) shows our reconstruction results. Notice that the left arm of the model that is undergoing motion is accurately reconstructed. There is some loss of detail in the reconstruction primarily due to limited number of views (twenty) and since we did not use 3D calibration (monocular un-calibrated camera sequence), performing visual hull intersection on cross-sectional slices using planar homographies. Yet to qualitatively assess the accuracy of our results we show in figure 5.7 (b) the reconstruction if the object is assumed to be rigid during the sequence and our occupancy deblurring approach is not used. It can be clearly

seen that the left arm of the model is carved out by the visual hull intersection due to it's motion.

5.3.1 Quantitative Analysis

To quantitatively analyze our algorithm we conducted an experiment in which we obtained several monocular sequences of an object. In each flyby of the camera the object was kept stationary but after each cycle the posture of the object changed a little. We call this the 'Superman' dataset and is shown in figure 5.8(a). It consists of seven flybys of the camera around the object as the object deforms (both arms moving) after each sequence but is rigid within each. We call these the *rigid sequences* and each consists of 14 views of the object at a resolution of 480x720 with the object occupying a region of approximately 150x150 pixels. Figure 5.8(a) shows three of the seven rigid sequences. Notice the changing postures of the object between sequences. This data was used to obtain seven rigid reconstructions of the object, three of which are shown in figure 5.9(a).

A monocular sequence of a non-rigidly deforming object was assembled by selecting two views from each rigid sequence in order, thereby creating a set of fourteen views of the object as it changes posture (deforms non-rigidily). Reconstruction using our occupancy deblurring approach was performed and the visualization of the results are shown in figure 5.9(b). Notice using our approach the arms of the object are accurately reconstructed which are carved out when traditional visual hull intersection is used as shown in figure 5.9(c). For a quantitative analysis we compared our reconstruction results with each of the seven

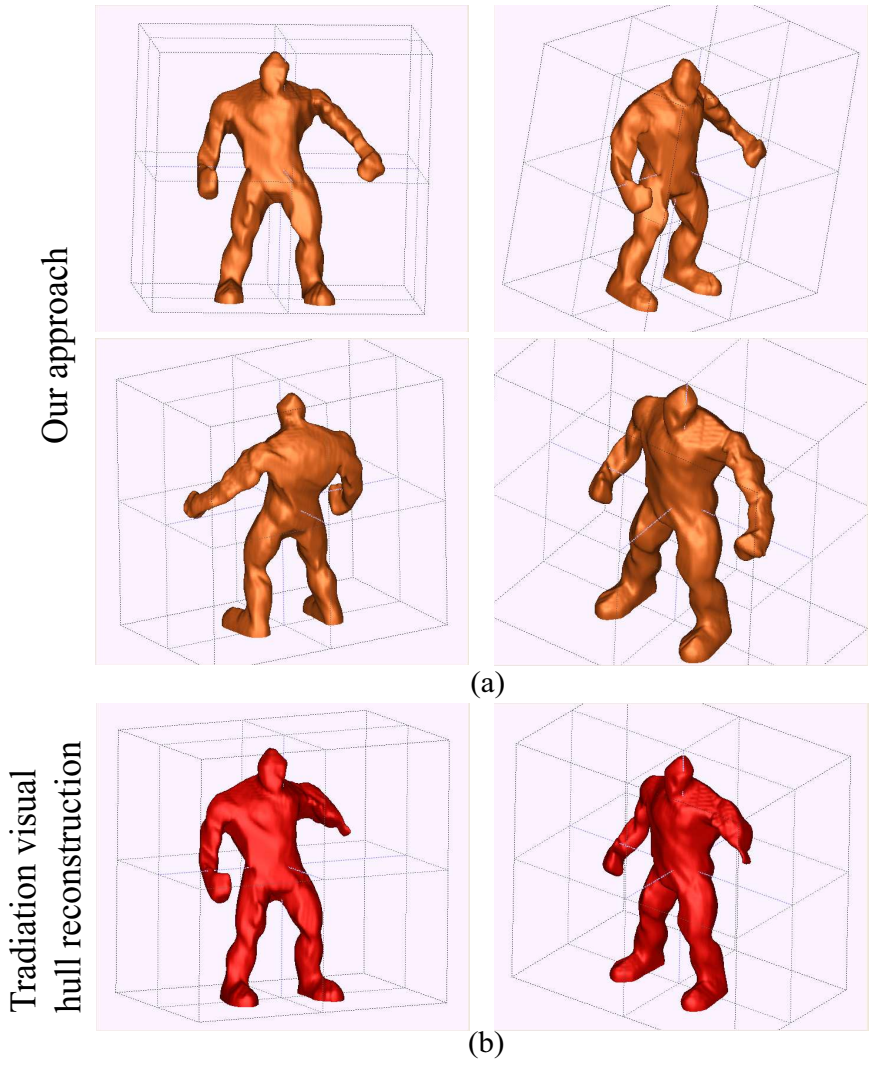


Figure 5.7: (a) Different views of the final reconstruction of the object dataset shown in figure 5.4. Notice how the left arm of the model that undergoes a non-rigid motion is accurately reconstructed. (b) Shows the reconstruction if a conventional visual hull intersection approach is used on the same data. The arm is carved out due to the motion.

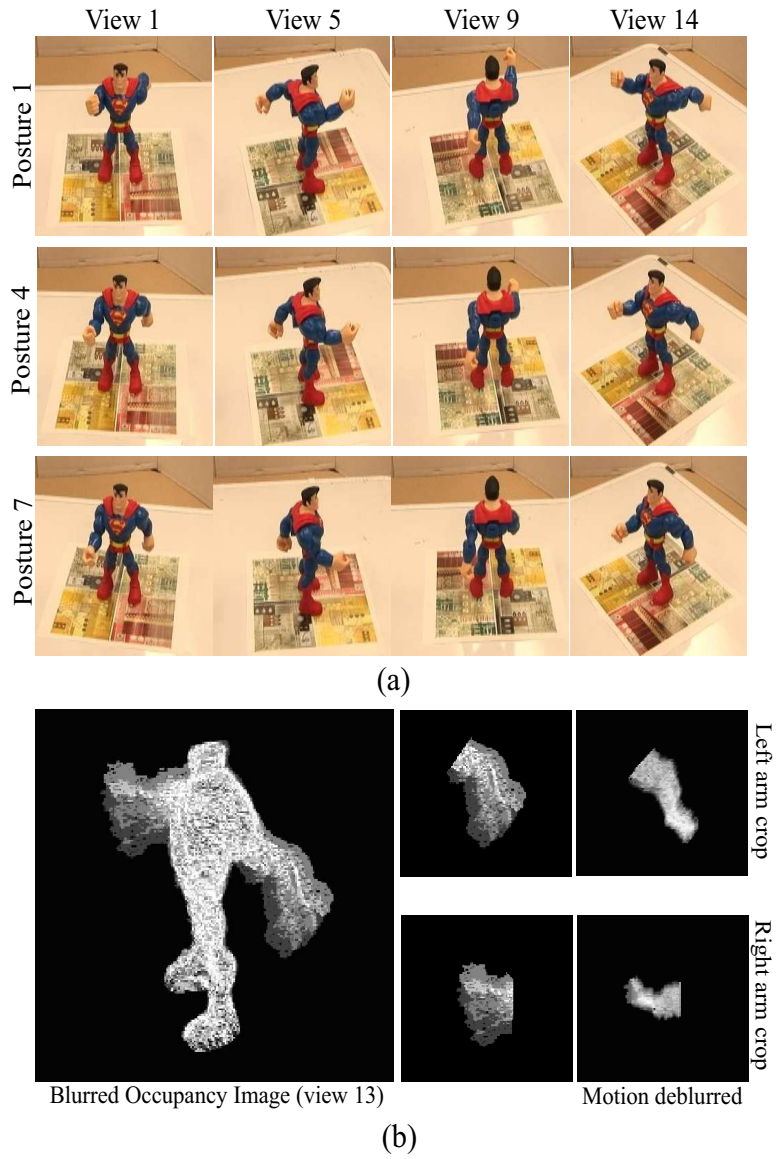


Figure 5.8: (a) Each row shows four views (of fourteen) from one of the seven monocular sequences in the dataset. The object is rigid within each sequence but changes posture between sequences by moving the arms (notice both arms moving progressively inwards). A monocular sequence of a non-rigidly deforming object is assembled by selecting two views in order from each rigid sequence. (b) The blurred occupancy image (one of fourteen) produced using our approach. The cropped, detail sections on the arms are shown on the right together with the deblurred results.

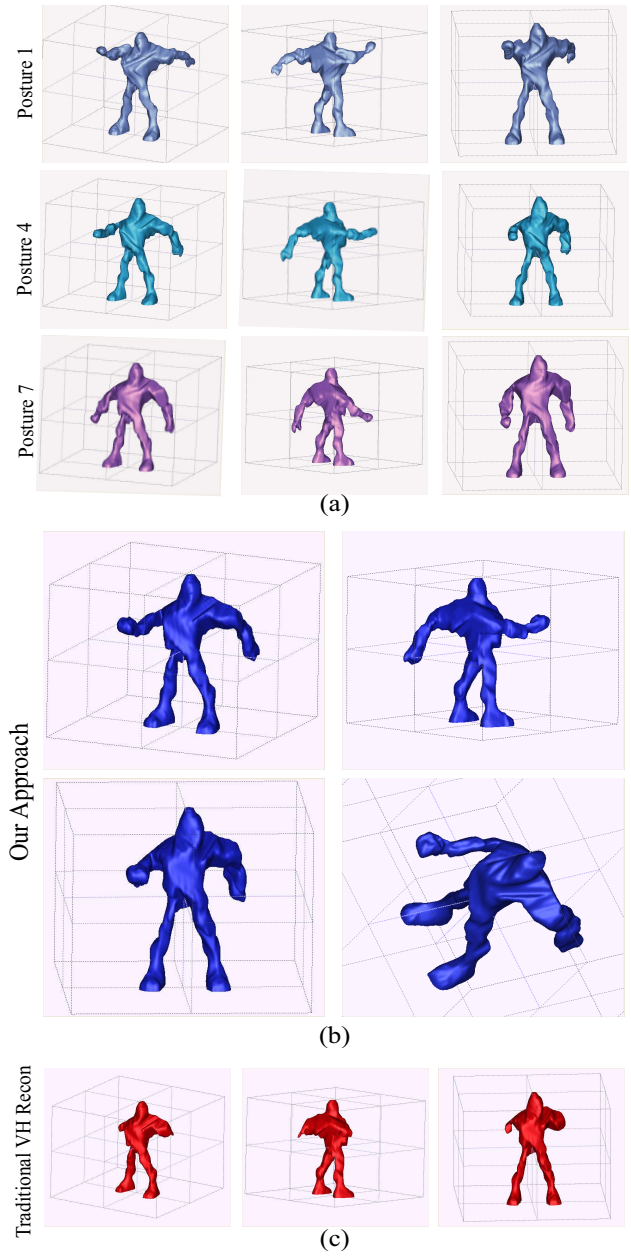


Figure 5.9: (a) Three of the seven visual hull reconstructions from the seven rigid sequences shown in figure 5.8(a). (b) Visualization of the reconstruction using our occupancy deblurring approach on the assembled non-rigid monocular sequence. Notice that the moving arms are accurately reconstructed using our approach but are carved out if we use conventional visual hull intersection that assumes the object is rigid as can be seen in the visualization in(c).

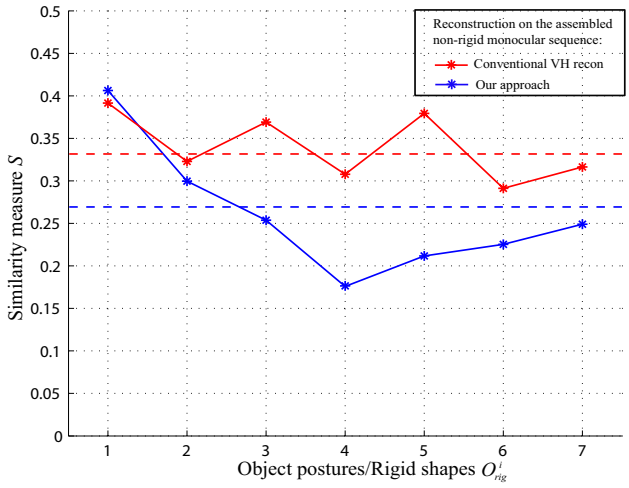


Figure 5.10: Plot of the similarity measure between reconstructions from the assembled monocular sequence and the rigid sequences in the 'Superman' dataset.

reconstructions from the rigid sequences. All the reconstructions were aligned in 3D (w.r.t the ground plane coordinate system) and the similarity was evaluated using a measure of the ratio of overlapping and non-overlapping voxels in the 3D shapes. The similarity measure is described as:

$$S_i = \left(\frac{\sum_{\forall v \in \mathbb{R}^3} \left((v \in O_{test}) \oplus (v \in O^i_{rig}) \right)}{\sum_{\forall v \in \mathbb{R}^3} \left((v \in O_{test}) \wedge (v \in O^i_{rig}) \right)} \right)^2, \tag{5.3}$$

where v is a voxel in the voxel space $\mathbb{R}^3$, O_{test} is the 3D reconstruction that needs to be compared with, O^i_{rig} the visual hull reconstruction from ith rigid sequence and S_i is the similarity score i.e. the square of the fraction of non-overlapping to overlapping voxels that are a part of the reconstructions. The closer S_i is to zero the greater the similarity.

In figure 5.10 we show a plot of the similarity measure computed by comparing the reconstructions from the assembled non-rigid monocular sequence using our deblurring based

approach (blue plot) and conventional visual hull intersection that assumes the object is rigid (red plot). Note for the red plot the similarity is consistently quite low (measure high). This is expected since the moving parts of the object (arms) are carved out by the visual hull intersection as can be seen in the visualization of this reconstruction in figure 5.9(c). For the blue plot notice a clear dip in the similarity measure value at rigid shape 4 demonstrating quantitatively that the result of using our approach is very similar to this shape. This also corroborates what can be visually observed by comparing our reconstruction results shown in figure 5.9(b) with the reconstruction of the fourth rigid sequence shown in the second row of figure 5.9(a).

5.4 Summary

In this chapter we have presented an image-based approach to reconstruct non-stationary, articulated objects from silhouettes obtained with a monocular video sequence. Our approach starts with an silhouette fusion step that combines color and silhouette images to produce occupancy likelihood maps in each view, where the values at each pixel correspond to the fraction of the total time duration that the pixel observed an occupied scene location. We call these the set of blurred occupancy images, analogous to a motion blurred image but in the scene occupancy domain. We then use a motion de-blurring approach to de-blur the occupancy images. The de-blurred occupancy images correspond to a silhouettes of the mean object shape during the motion and are used to obtain a visual hull reconstruction of the object. We have shown compelling results on challenging monocular datasets of rigid

and non-rigid articulated motion where traditional visual hull intersection approaches fail to reconstruct the object correctly.

In the next chapter we delve into a slightly different topic of object class detection. We show how our approach to 3D reconstruction can be applied to create rich and powerful models to address the challenging problem of object class detection from arbitrary view point.

CHAPTER 6
OBJECT CLASS DETECTION FROM ARBITRARY VIEW

6.1 Introduction

In recent years, the problem of object detection has received considerable attention from both the computer vision and machine learning communities. The key challenge of this problem is the ability to recognize any member in a category of objects in spite of wide variations in visual appearance due to geometrical transformations, change in viewpoint, or illumination.

As the approaches for recognizing an object class from some particular viewpoint or detecting a specific object from an arbitrary view are advancing toward maturity [143, 159, 161], solutions to the problem of object class detection using multiple views are still relatively far behind. Object detection can be considered even more difficult than classification, since it is expected to provide accurate location and size of the object.

Researchers in computer vision have studied the problem of multi-view object class detection resulting successful approaches following two major directions. One path attempts to use increasing number of local features by applying multiple feature detectors simultaneously [140, 148, 163, 164, 165]. It has been shown that the recognition performance can be benefited by providing more feature support. However, the spatial connections of the features in each view and/or between different views have not been pursued in these works.

These connections can be crucial in object class detection tasks. Recently, much attention has been drawn to the second direction related to multiple views for object class detection [146, 149, 150]. The early methods apply several single-view detectors independently and combine their responses via some arbitration logic. Features are shared among the different single-view detectors to limit the computational overload. Most recently, Thomas et al. [166] developed a single integrated multi-view detector that accumulates evidence from different training views. Their work combines a multi-view specific object recognition system [159], and the Implicit Shape Model for object class detection [161], where single-view codebooks are strongly connected by the exchange of information via sophisticated activation links between each other.

In this chapter we present a novel *3D feature model* based object class detection method to deal with these challenges. The objective of this work is to detect the object given an arbitrary 2D view using a general 3D feature model of the class. In our work, the objects can be arbitrarily transformed (with translation and rotation), and the viewing position and orientation of the camera is arbitrary as well. In addition, camera parameters are assumed to be unknown.

Object detection in such a setting has been considered a very challenging problem due to various difficulties of geometrically modeling relevant 3D object shapes and the effects of perspective projection. In this work, we exploit our homographic framework for 3D object shape reconstruction presented in previous chapters. Given a set of 2D images of an object taken from different viewpoints around the object with unknown camera parameters,

which are called *model views*, the 3D shape of this specific object is reconstructed using the homographic framework presented earlier in chapter 4 [160]. In our work, once the object is segmented from the occupancy data, the 3D shape is represented by a volume consisting of binary slices with 1 denoting the object and 0 for background. By using this method, we can not only reconstruct 3D shapes for the objects to be detected, but also have access to the homographies between the 2D views and the 3D models, which are then used to build the 3D feature model for object class detection.

In the feature modeling phase of our method, SIFT features [162] are computed for each of the 2D model views and mapped to the surface of the 3D model. Since it is difficult to accurately relate 2D coordinates to a 3D model by projecting the 3D model to a 2D view (with unknown camera parameters), we propose to use a homography transformation based algorithm. Since the homographies have been obtained during the 3D shape reconstruction process, the projection of a 3D model can be easily computed by integrating the transformations of slices from the model to a particular view, as opposed to directly projecting the entire model by estimation of the projection matrix. To generalize the model for object class detection, images of other objects of the class are used as *supplemental views*. Features from these views are mapped to the 3D model in the same way as for those model views. A codebook is constructed from all of these features and then a 3D feature model is built. The 3D feature model thus combines the 3D shape information and appearance features for robust object class detection.

Given a new 2D test image, correspondences between the 3D feature model and this testing view are identified by matching feature. Based on the 3D locations of the corresponding features, several hypotheses of viewing planes can be made. For each hypothesis, the feature points are projected to the viewing plane and aligned with the features in the 2D testing view. A confidence is assigned to each hypothesis and the one with the highest confidence is then used to produce the object detection result.

6.2 3D Feature Model Description and Training

In our method, not only the 3D shape of the target object is exploited, but also the appearance features. We relate the features with the 3D model to construct a *feature model* for object class detection.

6.2.1 Attaching 2D Features to 3D Model

The features used in our work are computed using the SIFT feature detector [162]. Feature vectors are computed for all of the training images. In order to efficiently relate the features computed from different views and different objects, all the detected features are attached to the 3D surface of the previously built model. By using the 3D feature model, we avoid storing all the 2D training views, thus there is no need to build complicated connections between the views. The spatial relationship between the feature points from different views are readily available, which can be easily retrieved when matched feature points are found.

The features computed in 2D images are attached to the 3D model by using our homographic framework described in chapters 3 and 4. Instead of directly finding the 3D location of each 2D feature, we map the 3D points from the model's surface to the 2D views, and find the corresponding features. Our method does not require the estimation of a projection matrix from 3D model to a 2D image plane, which is a non-trivial problem. In our work, the problem is successfully solved by transforming the model to various image planes using homography. Since the homographies between the model and the image planes have already been obtained during the construction of the 3D model, we are able to map the 3D points to 2D planes using homography transformation.

In our work, a 3D shape is represented by a binary volume V, which consists of K slices S_j, $j \in [1, K]$. As shown in Fig. 4.1, each slice of the object is transformed to a 2D image plane by using the corresponding homography in (4.8). The transformed slice accounts for a small patch of the object projection. Integrating all these K patches together, the whole projection of 3D object in the 2D image plane can be produced. In this way, we obtain the model projection by using a series of simple homography transformations and the hard problem of estimating the projection matrix of a 3D model to a 2D view is avoided. In our method, the 3D shapes are represented using binary volumes with a stack of slices along the reference direction. Thus, the surface points can be easily obtained by applying edge detection techniques. After transforming the surface points to 2D planes, feature vectors computed in 2D can be related to the 3D points according to their locations.

Figure 6.1: Construction of 3D feature model for motorbikes. 3D shape model of motorbike (at center) is constructed using the model views (images on the inner circle) taken around the object from different viewpoints. Supplemental images (outer circle) of different motorbikes are obtained by using Google's image search. The supplemental images are aligned with the model views for feature mapping. Feature vectors are computed from all the training images and then attached to the 3D model surface by using the homography transformation.

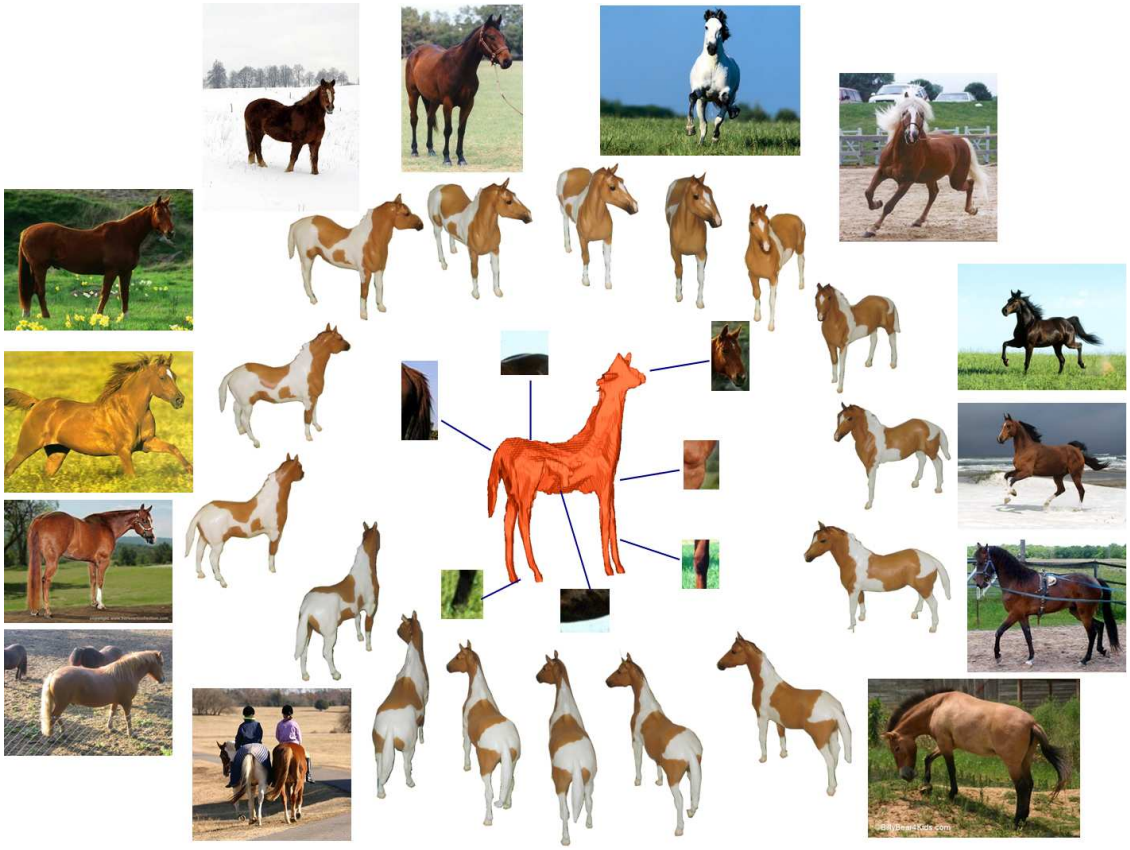


Figure 6.2: Construction of 3D feature model for horses. 3D shape model of horse (at center) is constructed using the model views (images on the inner circle) taken around a canonical toy horse from different viewpoints. Supplemental images (outer circle) of different horses are obtained by using Google's image search. The supplemental images are aligned with the model views for feature mapping. Feature vectors are computed from all the training images and then attached to the 3D model surface by using the homography transformation.

6.2.2 Beyond the Model Views

The training images in our work come from two sources. One set of images is taken around a specific object of the target class to reconstruct it in 3D as shown in Fig. 6.1. These images are called *model views*, which provide multiple views of the object but are limited to the specific object. To generalize the model for recognizing other objects in the same class, another set of training images is obtained by using Google image search. Images of objects in the same class with different appearances and postures are selected. These images are denoted as the *supplemental views*. By augmenting the 3D model with appearance features from supplemental views we are creating a rich model that captures not only the varying appearance of the object across different instances but also the 3D spatial arrangement of these features.

Since the homographies between the supplemental images and the 3D model are unknown, features computed from the supplemental images cannot be directly attached to the feature model. Instead, we utilize the model views as bridges to connect the supplemental images to the model as illustrated in Fig. 6.1. In the training phase for each supplemental image, the model view, which has the most similar viewpoint is manually specified. An affine transform is computed between the object bounding boxes in the two views. The supplemental image is then warped with this affine transformation and cropped/padded to the dimensions of the selected model view. This perturbation of the supplemental image allows us to relate it to the object 3D model using the same pencil of homographies that relate the corresponding model view. This process is illustrated in figure 6.3. By repeating the process for each

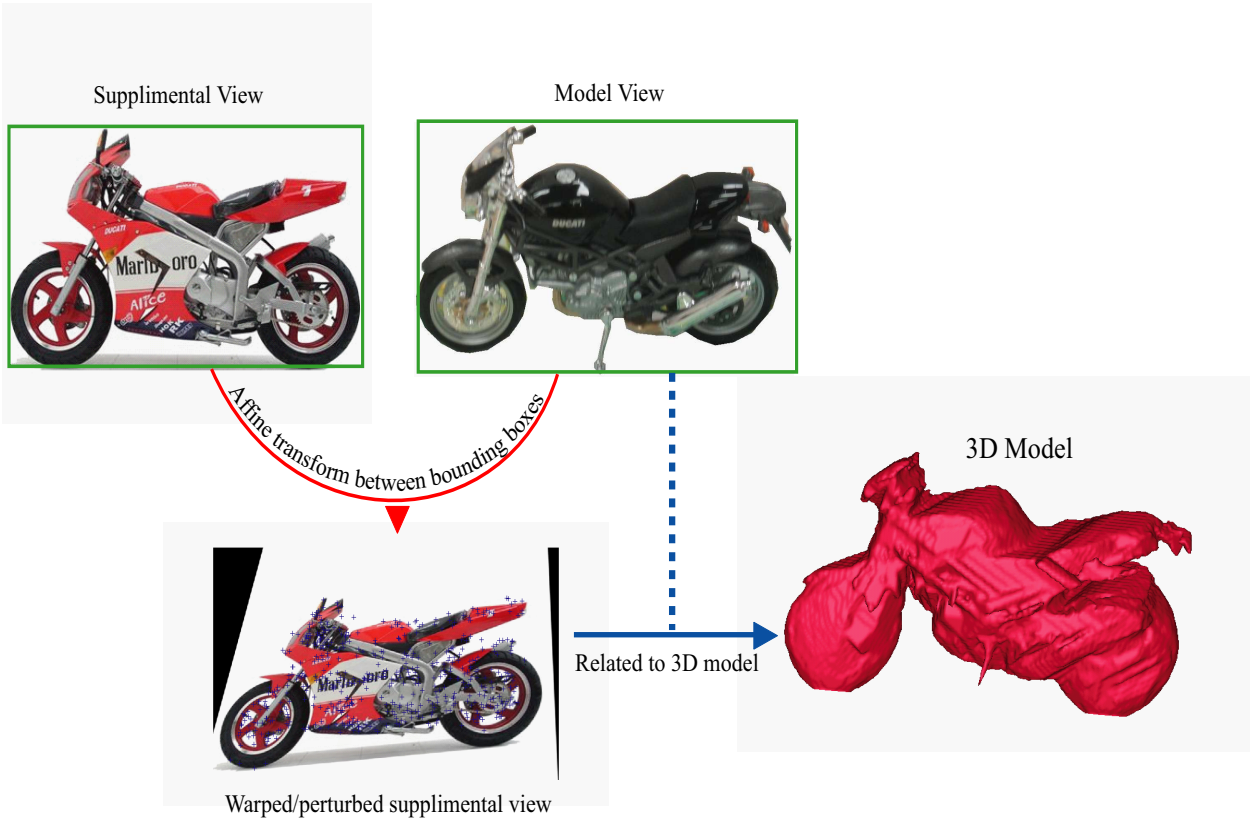


Figure 6.3: In the training phase for each supplemental view we specify the closest model view. An affine transform is computed between the bounding boxes of the object in the supplemental and the model views. The supplemental view is warped/perturbed with this affine transform as shown in the figure and it's dimensions are cropped or padded accordingly. The perturbed supplemental view is then linked to the 3D model using the same pencil of homographies that link the selected model view to the 3D model.

supplemental view 2D features computed from all of the supplemental training images can now be correctly attached to the 3D model surface. A codebook is constructed by combining all the mapped features with their 3D locations.

It should be noted that the 3D feature model will be most effective and powerful if a large number of supplemental views are used to augment the core 3D model. A small number of supplemental views may in fact only serve to throw off the 3D feature model from recognizing the original training views. But with increasing density of features from a large number of supplemental views, the 3D feature model will tend to approach a canonical representation of the object class. For instance in creating a 3D feature model of a motor bike (figure 6.1) appearance features from the steering sections of the motor bike training views will tend to cluster up in the steering parts of the 3D model and likewise for other parts of the motor bike.

6.3 Object Class Detection

Given a new test image, our objective is to detect objects belonging to the same class in this image by using the learnt 3D feature model M. Each entry of M consists of a code and its 3D locations $\{c, l_c^3\}$. Let F denote the SIFT features computed from the input image, which is composed by the feature descriptor and its 2D location in the image $\{f, l_f^2\}$. Object O_n is detected by matching the features F to the 3D feature model M.

In our work, feature matching is achieved in three phases. In the first phase, we match the features by comparing all the input features to the codebook entries in Euclidean space.

However, not all the matched codebook entries in 3D are visible at the same time from a particular viewpoint. So, in the second phase, matched codes in 3D are projected to viewing planes and hypotheses of viewpoints are made by selecting viewing planes with the largest number of visible points projected. In the third phase, for each hypothesis, the projected points are compared to 2D matched feature points using both feature descriptors and locations. This is done by iteratively estimating the affine transformation between the feature point sets and removing the outliers with large distance between corresponding points. Outliers belonging to the background can be rejected during this matching process. The object location and bounding box is then determined according to the 2D locations of the final matched feature points. The confidence of detection is given by the degree of match.

6.4 Experimental Results

The proposed method has been tested on two object classes: motorbikes and horses. For the motorbikes, we took 23 model views around a motorbike and obtained 45 supplemental views by using Google's image search. Some training images of the motorbikes and the 3D shape model are shown in Fig. 6.1. For the horses, 18 model views were taken and 51 supplemental views were obtained.

To measure the performance of our 3D feature model based object class detection technique, we have evaluated the method on the PASCAL VOC Challenge 2006 test dataset [145], which has become a standard testing dataset for objective evaluation of object classification and detection algorithms. The dataset is very challenging due to the large variability

in the scale and poses, the extensive clutter, and poor imaging conditions. Some successful detection results are shown in Figures 6.4 and 6.5. The green box indicates the ground truth, while our results are shown in red boxes.

For quantitative evaluation, we adopt the same evaluation criteria used in PASCAL VOC challenge, so that our results can be directly comparable with [150, 166, 145]. By using this criteria, a detection is considered correct, if the area of overlap between the predicted bounding box B_p and ground truth bounding box B_{gt} exceeds 50% using the formula

$$\frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} > 0.5. \tag{6.1}$$

The *average precision* (AP) and *precision-recall* (PR) curve can then be computed for performance evaluation.

Fig. 6.6(a) shows the PR curves of our approach and the methods in [150, 166] for motorbike detection. The curve of our approach shows a substantial improvement over the precision compared to the method in [150], which is also indicated by the AP value (0.182). Although our performance is lower than that of [166], considering the smaller training image set used in our experiments, this can be regarded as satisfactory. Fig. 6.6(b) shows the performance curves for horse detection. While there is no result reported in the VOC challenge using researchers' own training dataset for this task, we compared our result to those using the provided training dataset. Our approach performs better than the reported methods and obtained AP value of 0.144. It is noted that the absolute performance

Figure 6.4: Detection of motorbikes in the PASCAL VOC dataset using our approach. The ground truth is shown in green and red boxes display our detected results.

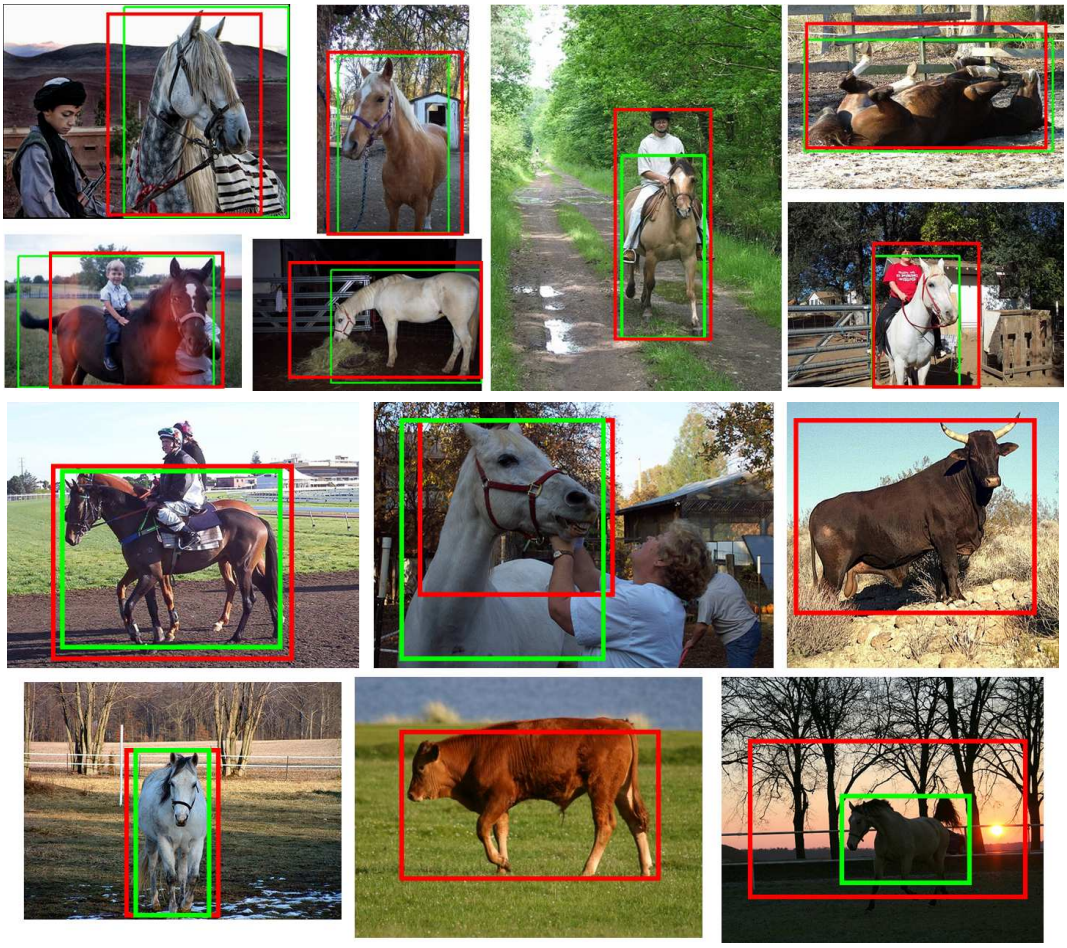


Figure 6.5: Detection of horses in the PASCAL VOC dataset using our approach. The ground truth is shown in green and red boxes display our detected results.

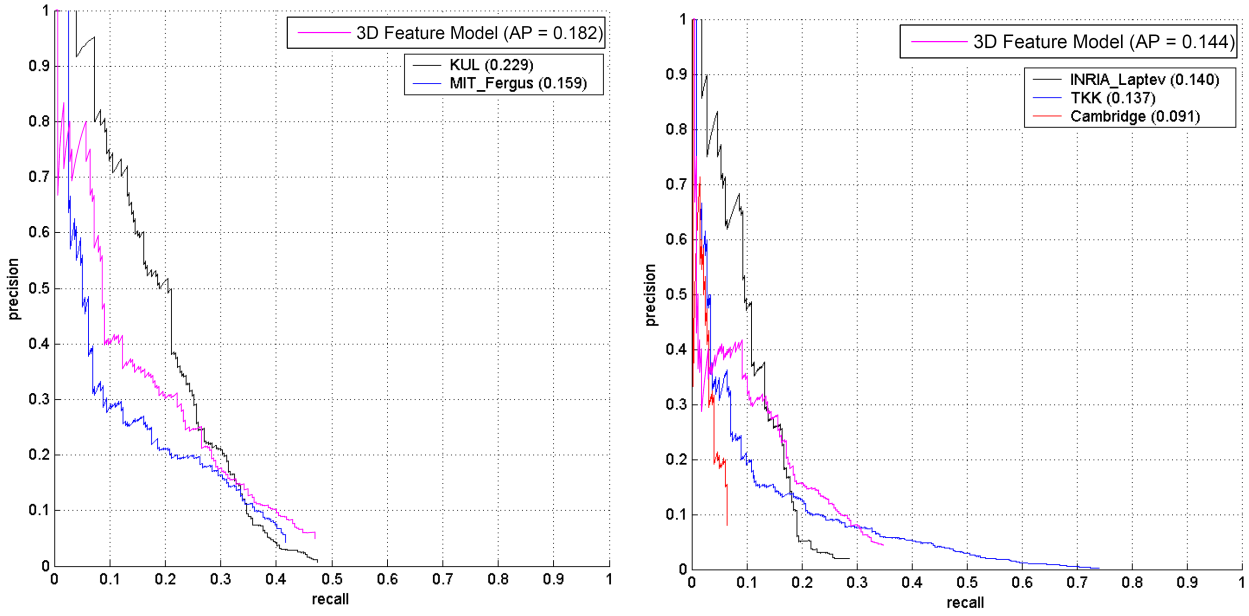


Figure 6.6: The PR curves for (a) motorbike detection and (b) horse detection using our 3D feature model based approach. The curves reported in [145] on the same test dataset are also included for comparison.

level is lower than that of motorbike detection, which might be caused by the non-rigid body deformation of horses.

6.5 Summary

In this chapter, we have proposed a multi-view object class detection method based on 3D object shape and appearance modeling. We develop a 3D feature model for establishing spatial connections between different views by mapping appearance features to the surface of a 3D shape. This is achieved using the homographic framework for 3D modelling presented in chapter 4. Experimental evaluation of the proposed method suggests collaborative

information in the 2D training images can be represented in a more unified way through a 3D feature model of the object. We have also revealed that both appearance and shape can be salient properties to assist in object detection. Performance of the proposed method has been evaluated using the PASCAL VOC challenge dataset and promising results have been demonstrated. In our future work, we plan to extend our method by taking supplemental views in a more automated fashion. So, more supplemental views can be easily incorporated to improve the performance.

CHAPTER 7
CONCLUSIONS

In this thesis we have presented novel multi-view approaches for the tracking, 3D reconstruction and detection of scene objects. First we have presented an algorithm that can reliably track multiple people in a complex environment. This is achieved by resolving occlusions and localizing people on scene planes using a planar homographic occupancy constraint. Combining foreground likelihood information from multiple views and obtaining the global optimum of space-time scene occupancies over a window of frames we segment out the individual trajectories of the people. We have presented detailed quantitative results on challenging multi-view datasets.

Second we have presented an image-based visual hull approach for fusing foreground silhouette information from multiple views. Unlike other visual hull based methods that require calibrated views and use 3D constructs like voxels, 3D visual cones or polygonal meshes, our method uses only minimal geometric information i.e. homographies between views and the vanishing points of a reference direction. We perform visual hull intersection *in* the image plane without without requiring to go in 3D space. This process is extended to multiple planes parallel to the reference plane, each computation delivering a successive cross-sectional slice of the object.

We have extending our 3D reconstruction algorithm to monocular video sequences of non-stationary, articulated objects. We introduce the concept of *motion blurred scene occupancies*, a direct analogy of motion blurred images but in a 3D object scene occupancy space resulting from the motion/deformation of the object. The de-blurred occupancy information corresponds to silhouettes of a mean/motion compensated object shape and are used to obtain a visual hull reconstruction of the object.

Finally, we have presented an object class detection method based on 3D object modeling. Instead of using a complicated mechanism for relating multiple 2D training views, our method establishes spatial connections between these views by mapping them directly to the surface of 3D model. Features are computed in each 2D model view and mapped to the 3D shape model. To generalize the model for object class detection, features from supplemental views are also considered. A codebook is constructed from all of these features and then a 3D feature model is built. Given a 2D test image, correspondences between the 3D feature model and the testing view are identified by matching the detected features.

CHAPTER 8
DIRECTIONS FOR FUTURE WORK

There are many possible extensions of the work presented in this thesis. One direction is to incorporate color models in the detection and tracking of individual people. The color models can be used to disambiguate tracks in cases when two or more people come too close to be segmented as separate entities. Using articulated human shape models can be another addition that can act as a prior to prune out false detections and increase robustness of localization. Similarly a human motion model that takes into account the consistency of speed and direction as well as modelling collision avoidance strategies between people could be an interesting addition. These models may be useful in situations where crowd densities increase and camera views are limited. The tracking results can also be used in the analysis of group or crowd activities. People in crowds tend to show peculiar patterns of collective behavior like, gathering, scattering, clique behavior, marching, milling etc. An interesting direction of research would be to develop an ontology of collective pedestrian behaviors in crowds and use the tracking results to detect and recognize both normal and abnormal activities.

Many new ideas and applications can also be explored using our 3D reconstruction approach. Different modalities, like infra-red imagery, can be seamlessly integrated into our method. Human body pose estimation and marker-less motion capture are a direct appli-

cation of our approach and may benefit from the relaxation of complete camera calibration. In relation to our work on reconstructing non-stationary articulated objects in monocular video, one major direction of research is to handle non-uniform motion blur in occupancy images. Currently in our implementation, the user specifies sections of uniform motion blur in the blurred occupancy images, which are then restored independently. In future studies this step may be automated or the focus can be on developing elegant solutions to non-uniform motion de-blurring.

In the context of our approach to object class detection one important direction of future studies will be to automate the matching of supplemental views to model views in the training phase. Currently in the training phase user specifies the closest model view to perturb the supplemental views and integrate them into building the 3D feature model. Though this process is acceptable in creating training data, it tends to become cumbersome with a large number of supplemental views. Another interesting direction for future work is to extend our 3D feature model for the analysis of human actions from arbitrary viewpoints.

LIST OF REFERENCES

[1] S. M. Khan, M. Shah. A multi-view approach to tracking people in crowded scenes using a planar homography constraint. In European Conference on Computer Vision, 2006.

[2] S. M. Khan, P. Yan, M. Shah. A Homographic Framework for the Fusion of Multi-view Silhouettes. To appear in proceedings of International Conference of Computer Vision, 2007.

[3] P. Yan, S. M. Khan and M. Shah. 3D Model Based Object Recognition from Arbitrary View, IEEE International Conference of Computer Vision (ICCV), Brazil, 2007.

[4] S. M. Khan and M. Shah. Reconstructing Non-stationary Articulated Objects in Monocular Video using Silhouette Information, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Alaska, 2008.

[5] M. Han, W. Xu, H. Tao, and Y. Gong. An algorithm for multiple object trajectory tracking. In Conference on Computer Vision and Pattern Recognition, volume 1, pages 864871, June 2004.

[6] T. Zhao and R. Nevatia. Tracking multiple humans in complex situations. IEEE Trans. Pattern Anal. Mach. Intell., 26(9):12081221, 2004.

[7] T. Zhao and R. Nevatia, Tracking multiple humans in crowded environment. IEEE Conference on Computer Vision and Pattern Recognition, 2004.

[8] M. Isard and J. MacCormick. Bramble: a bayesian multiple-blob tracker. In Conference on Computer Vision and Pattern Recognition, volume 2, pages 3441, July 2001.

[9] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe. A boosted particle filter: multitarget detection and tracking. In ECCV, Prague, Czech Republic, May 2004.

[10] G.J. Brostow and R. Cipolla. Unsupervised Bayesian Detection of Independent Motion in Crowds. IEEE CVPR 2006.

[11] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In CVPR (1), pages 878885, 2005.

[12] Z. Khan, T. R. Balch, and F. Dellaert. An mcmc-based particle filter for tracking multiple interacting targets. In ECCV (4), pages 279290, 2004.

[13] S.J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. Tracking Groups of People, CVIU 2000.

[14] R. Rosales and S. Sclaroff. 3D Trajectory Recovery for Tracking Multiple Objects and Trajectory Guided Recognition of Actions, CVPR 1999.

[15] H. Sidenbladh, M.J. Black, D.J. Fleet. Stochastic Tracking of 3D Human Figures Using 2D Image Motion, ECCV 2000.

[16] I. Haritaoglu, D. Harwood, and L. Davis. Who, when, where, what: A real time system for detecting and tracking people. In FGR, pages 222227, 1998.

[17] I.K. Sethi and R. Jain, Finding tracjectories of feature poins in a monocular image sequence, IEEE PAMI, 1987.

[18] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. IJCV, vol. 39, no. 1, pp. 57-71, 2000.

[19] A. Yilmaz, X. Li, and M. Shah. Contour-Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No.11, pp. 1531-1536, 2004.

[20] A. Yilmaz, O. Javed, and M. Shah, Object Tracking: A Survey, ACM Journal of Computing Surveys, Vol. 38, No. 4, 2006.

[21] Y. Huang, I. Essa. Tracking Multiple Objects Through Occlusions. IEEE CVPR 2005.

[22] Y. Wu, T. Yu, and G. Hua. Tracking appearances with occlusions. In CVPR, 2003.

[23] D. Koller, J. Weber, and J. Malik. Robust multiple car tracking with occlusion reasoning. In European Conference on Computer Vision, pages 189196, 1994.

[24] A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, and R. Bolle. Appearance models for occlusion handling. In IEEE Workshop on Performance Evaluation of Tracking and Surveillance, 2001.

[25] N. Jojic and B.J. Frey. Learning flexible sprites in video layers. In IEEE Conference on Computer Vision and Pattern Recognition, 2001.

[26] H. Tao, H. S. Sawhney, and R. Kumar. Object tracking with bayesian estimation of dynamic layer representations. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(1):7589, 2002.

[27] A.G. Perera, C. Srinivas, A. Hoogs, G. Brooksby, W. Hu. Multi-Object Tracking Through Simultaneous Long Occlusions and Split and Merge Conditions. CVPR 2006.

[28] P. Kelly, A. Katkere, D. Kuramura, S. Moezzi, S. Chatterjee, R. Jain, An Architecture for Multiple Perspective Interactive Video, ACM Proceedings of the Conference on Multimedia, 1995.

[29] K. Sato, T. Maeda, H. Kato, and S. Inokuchi, CAD-Based Object Tracking With Distributed Monocular Camera For Security Monitoring, Proceedings of IEEE Workshop on CAD-Based Vision, 1994.

[30] A. Nakazawa, H. Kato, S. Inokuchi, Human Tracking Using Distributed Vision Systems, Proceedings of the International Conference on Pattern Recognition, 1998.

[31] R. Jain and K. Wakimoto, Multiple Perspective Interactive Video, IEEE International Conference on Multimedia Computing and Systems, 1995.

[32] Orwell, J., Massey, S., Remagnino, P., Greenhill, D., and Jones, G.A. 1999. A Multi-agent framework for visual surveillance, ICIP 1999.

[33] Cai, Q. and Aggarwal, J.K. 1998. Automatic tracking of human motion in indoor scenes across multiple synchronized video streams, ICCV 1998.

[34] J. Kang, I. Cohen and G. Medioni, Continuous Tracking Within and Across Camera Streams, IEEE Conference on Computer Vision and Pattern Recognition, 2003.

[35] S. Dockstader and A. Tekalp, Multiple Camera Fusion for Multi-Object Tracking, IEEE International Workshop on Multi Object Tracking.

[36] T.H. Chang and S. Gong, Tracking Multiple People with a Multi-Camera System, IEEE Workshop on Multi-Object Tracking, 2001.

[37] T. Darrell, D. Demirdjian, N. Checka and P. Felzenszwalb, Plan-view Tracjectory Estimation with Dense Stereo Background Models, IEEE International Conference on Computer Vision, 2001.

[38] Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., and Shafer, S. 2000. Multi-camera multi-person tracking for easy living, IEEE International Workshop on Visual Surveillance.

[39] Mittal, A., Larry, S.D. M2Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene. IJCV, 2002.

[40] B. Leibe, N. Cornelis, K. Cornelis, and L. van Gool, Dynamic 3D Scene Analysis from a Moving Vehicle. IEEE Computer Vision and Pattern Recognition, 2007.

[41] J. Franco, E. Boyer. Fusion of Multi-View Silhouette Cues Using a Space Occupancy Grid, ICCV 2005.

[42] A. Elfes. Occupancy grids: a probabilistic framework for robot perception and navigation. PhD thesis, 1989.

[43] S. Thrun. Learning Occupancy Grid Maps with Forward Sensor Models. Journal of Autonomous Robots, 2003.

[44] 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, 2003.

[45] A. Azarbayejani and A. Pentland, Real-Time Self-Calibrating Stereo Person Tracking Using 3D Shape Estimation from Blob Features, Proceedings on International Conference on Pattern Recognition, 1996.

[46] C.R. Wren, A. Azarbayejani, T. Darell, and A.P. Pentland, "Pfinder: Real-Time Tracking of Human Body," IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997.

[47] Stauffer, C., Grimson, W.E.L. 1999. Adaptive background mixture models for real-time tracking, CVPR 1999.

[48] S. Osher, and J. Sethian. Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations Journal of Computational Physics 1988.

[49] D.E. Schmieder, M.R. Weathersby, Performance Detection with Variable Clutter Resolution. IEEE Transactions of Aerospace and Electronic Systems, 1983.

[50] S.R. Rotman, G. Tidhar, and M.L. Kowalczyk, Clutter metrics for target detection systems. IEEE Transactions on Aerospace and Electronic Systems, 30, 1 (Jan. 1994).

[51] Irani, M., Rousso, B. and Peleg, S. 1994. Computing Occluding and Transparent Motions. IJCV, Vol. 12, No. 1.

[52] Gurdjos, P. and Sturm, P. Methods and Geometry for Plane-Based Self-Calibration. CVPR, 2003.

[53] A. Criminisi, I. Reid and A. Zisserman. Single View Metrology, International Journal of Computer Vision, 1999.

[54] D. G. Lowe. Distinctive Image Features from Scale Invariant Keypoints, International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.

[55] R. Hartley, A. Zisserman. Multiple View Geometry in Computer Vision, Cambridge University Press.

[56] C. Rother. A new approach for vanishing point detection in architectural environments. In BMVC, 2002.

[57] F. Lv, T. Zhao, R. Nevatia. Self-Calibration of a camera from video of a walking human. IEEE ICPR 2002.

[58] Gibson, J.J. The Ecological Approach to Visual Perception. Boston: Houghton Mifflen 1979.

[59] Marr, D. 1982. Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. New York: W.H. Freeman.

[60] D. Marr and T. Poggio. Cooperative computation of stereo disparity. Science, vol.194:283287, 1976.

[61] S.T. Barnard and M.A. Fischler. Computational stereo. In Comput. Surveys, volume 14(4), pages 553572, 1982.

[62] U.R. Dhond and J.K. Aggarwal. Structure from stereoa review. In IEEE Transactions on Systems, Man, and Cybernetics, volume 19(6), pages 14891510, December 1989.

[63] D. Marr and E. Hildreth. Theory of edge detection. Proceedings of the Royal Society of London, 207:187217, 1980.

[64] J. Mayhew and J. Frisby. Psychophysical and computational studies towards a theory of human stereopsis. Artificial Intelligence, 17:349385, 1981.

[65] J. Canny. A computational approach to edge detection. IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679698, 1986.

[66] R. Deriche. Using Cannys criteria to derive a recursively implemented optimal edge detector. Int. Journal of Computer Vision, 1(2):167187, 1987.

[67] S.B. Pollard, J.E.W. Mayhew, and J.P. Frisby. A stereo correspondence algorithm using a disparity gradient limit. In Perception, volume 14, pages 449470, 1985.

[68] W. Grimson. A computer implementation of a theory of human stereo vision. Philosoplical Transactions of the Royal Society of London., 292:217 253, 1981.

[69] N. Ayache and B. Faverjon. Efficient registration of stereo images by matching graph descriptions of edge segments. Int. Journal of Computer Vision, 1(2):107131, 1987.

[70] O. Faugeras and B. Mourrain, On the Geometry and Algebra of the Point and Line Correspondences Between N Images, IEEE ICCV 1995.

[71] R. Hartley, Estimation of relative camera positions for uncalibrated cameras, European Conference on Computer Vision, 1992.

[72] D. Freedman and T. Zhang. Interactive Graph Cut Based Segmentation With Shape Priors. IEEE CVPR 2005.

[73] H.H. Baker and T.O.Binford. Depth from edge and intensity based stereo. In Proc 7th Int. Joint Conf. Artificial Intelligence, pages 631636, August 1981.

[74] A. Rosenfeld, R. Hummel, and S. Zucker. Scene labeling by relaxation operations. 6:420433, June 1976.

[75] S. Roy and I.J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In Proc. 6th International Conference of Computer Vision, pages 492499, Bombay, India, January 1998. IEEE Computer Society Press.

[76] O. Veksler Y. Boykov and R. Zabih. Fast approximate energy minimisation via graph cuts. In Proc. 7th International Conference of Computer Vision, pages 377384, Kerkyra, Greece, September 1999. IEEE Computer Society Press.

[77] H. Ishikawa and D. Geiger. Occlusion, discontinuities, and epipolar lines in stereo. In H. Burkhardt and B. Neumann, editors, Proc. 5th European Conference on Computer Vision, volume 1406 of Lecture Notes in Computer Science, pages 232248, Freiburg, Germany, June 1998. Springer-Verlag.

[78] C. Dyer. Volumetric scene reconstruction from multiple views. In L.S. Davis, editor, Foundations of image analysis. Kluwer, Boston, 2001.

[79] R. Szeliski. Stereo algorithms and representations for image based rendering. In T. Pridmore and D. Elliman, editors, Proc. 10th British Machine Vision Conference, pages 314328, Nottingham, September 1999.

[80] P.A. Beardsley, P.H.S. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In Bernard Buxton and Roberto Cipolla, editors, Proc. 4th European Conference on Computer Vision, volume 1064 of Lecture Notes in Computer Science, pages 683695, Cambridge, UK, April 1996. Springer-Verlag.

[81] P.E. Debevec, C.J. Taylor, and J. Malik. Modelling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In Holly Rushmeier, editor, Proc. Special Interest Group on Computer Graphics, Annual Conference Series, pages 1120, New Orleans, LA, USA, August 1996. ACM SIGGRAPH, Addison Wesley.

[82] R. Szeliski S. Baker and P. Anandan. A layered approach to stereo reconstruction. In Proc. Computer Vision and Pattern Recognition, pages 434441, Santa Barbara, CA, USA, June 1998. IEEE Computer Society Press.

[83] A. Laurentini. The Visual Hull Concept for Silhouette- Based Image Understanding. IEEE TPAMI, 1994.

[84] J. S. Franco, E. Boyer. Fusion of Multi-View Silhouette Cues Using a Space Occupance Grid. IEEE ICCV, 2005.

[85] A. Yezzi and S. Soatto, Stereoscopic segmentation, IJCV, vol.53(1), pp. 31–43, 2003.

[86] R. Cipolla and D.P. Robertson. 3D models of architectural scenes from uncalibrated images and vanishing points. In IAPR 10th International Conference on Image Analysis and Processing, pages 824829, Venice, September 1999.

[87] R. Koch. 3-D surface reconstruction from stereoscopic image sequences. In Proc. 5th International Conference of Computer Vision, pages 109114, Cambridge, MA., USA, June 1995. IEEE Computer Society Press.

[88] R. Koch, M. Pollefeys, and L. van Gool. Multi viewpoint stereo from uncalibrated video sequences. In H. Burkhardt and B. Neumann, editors, Proc. 5th European Conference on Computer Vision, volume II of Lecture Notes in Computer Science, pages 5571, Freiburg, Germany, June 1998. Springer-Verlag.

[89] K. Kutulakos and S. Seitz. A Theory of Shape by Space Carving. IJCV, 38(3):199.218, 2000.

[90] D. Snow, P. Viola, and R. Zabih. Exact voxel occupancy with graph cuts. In CVPR, 2000.

[91] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, R. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. IEEE CVPR 2006.

[92] P.H.S. Torr, A. Dick, and R. Cipolla. Layer extraction with a Bayesian model of shapes. In D. Vernon, editor, Proc. 5th European Conference on Computer Vision, volume II of Lecture Notes in Computer Science, pages 273289, Dublin, Ireland, June 2000. Springer-Verlag.

[93] R. Szeliski and P. Golland. Stereo matching with transparency and matting. Int. Journal of Computer Vision, 32(1):4561, 1999.

[94] S.M. Seitz and C.M. Dyer. Photorealistic scene reconstruction by voxel coloring. Int. Journal of Computer Vision, 35(2):10671073, 1999.

[95] J.A. Sethian. Level set methods and fast marching methods. Cambridge University Press ISBN 0-521-64204-3, 2nd edition, 1999. Previously published as: Level Set Method

[96] Olivier Faugeras and Renaud Keriven. Variational principles, surface evolution, PDEs, level set methods, and the stereo problem. Transactions on Image Processing. Special Issue on Geometry Driven Diffusion and PDEs in Image Processing, 7(3):336344, March 1998.

[97] B. Culbertson, T. Malzbender, and G. Slabaugh. Generalized voxel coloring. In B. Triggs, A. Zisserman, and R. Szeliski, editors, Vision algorithms: Theory and Practice, volume 1883 of Lecture Notes in Computer Science, pages 100115, Kerkyra, Greece, September 1999. Springer-Verlag.

[98] E. Chen and L. Williams. View interpolation for image synthesis. Computer Graphics Proceedings, Annual Conference Series, pages 279288, 1993.

[99] S.M. Seitz and C.R. Dyer. Physically-valid view synthesis by image interpolation. Proc. IEEE Workshop on Representation of Visual Scenes, pages 1825, 1995.

[100] S.M. Seitz. Image-based transformation of viewpoint and scene appearance. PhD thesis, University of Wisconsin - Madison, 1997.

[101] S. Laveau and O. Faugeras. 3-D Scene representation as a collection of images and fundamental matrices. Technical Report No. 2205, INRIA, 1994.

[102] T.Werner, R.D. Hersch, and V. Hlavac. Rendering real-world objects using view interpolation. In Proc. 5th International Conference of Computer Vision, pages 957962, Cambridge, MA., USA, June 1995. IEEE Computer Society Press.

[103] L.N. Chang and A. Zakhor. View generation for three-dimensional scenes from video sequences. IEEE Trans. Image Processing, pages 584598, 1997.

[104] S. Avidan and A. Shashua. Novel view synthesis in tensor space. In Proc. Computer Vision and Pattern Recognition, pages 10341040, San Juan, Puerto Rico, June 1997. IEEE Computer Society Press.

[105] L. McMillan and G. Bishop. Plenoptic modelling: An image-based rendering system. In Proc. Special Interest Group on Computer Graphics, Annual Conference Series, pages 3946. ACM SIGGRAPH, Addison Wesley, August 1995.

[106] M. Levoy and P. Hanrahan. Light field rendering. pages 3142, August 1996.

[107] E.H. Adelson and J.R. Bergen. The plenoptic function and the elements of early vision. In M.S. Landy and J.A. Movshon, editors, Computational Models of Visual Processing., pages 320. MIT Press, 1991.

[108] Neisser, U. 1976. Cognition and Reality: Principles and Implications of Cognitive Psychology. San Francisco: W.H. Freeman.

[109] Poore, A.B. 1995 Multidimensional Assignments and Multitarget Tracking. Proc. Partitioning Data Sets; DIMACS Workshop 1995.

[110] Reid, D.B. 1979. An Algorithm for Tracking Multiple Targets. IEEE Trans. Automatic Control 1979.

[111] Y. Boukov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In Proc. ICCV, 2001.

[112] Y. Boykov, O. Veksler and R. Zabih. Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence, November2001.

[113] http://www.cs.cornell.edu/ rdz/graphcuts.html

[114] http://sceptre.king.ac.uk/sceptre/default.html

[115] A. Laurentini. The Visual Hull Concept for Silhouette- Based Image Understanding. IEEE TPAMI, 1994.

[116] G. K. M. Cheung, S. Baker, and T. Kanade. Visual hull alignment and refinement across time:a 3D reconstruction algorithm combining shape-frame-silhouette with stereo. In Proc. of CVPR03, June 2003.

[117] G. K. M. Cheung, S. Baker, and T. Kanade. Shape-From-Silhouette of Articulated Objects and its Use for Human Body Kinematics Estimation and Motion Capture. In Proc. of CVPR03, June 2003.

[118] G. P. Stein, A. Shashua. Model-Based Brightness Constraints: On Direct Estimation of Structure and Motion. IEEE TPAMI 2000.

[119] A. Yezzi and S. Soatto, Stereoscopic segmentation, IJCV, vol.53(1), pp. 31–43, 2003.

[120] S. Nayar and Y. Nakagawa. Shape from focus. IEEE Trans. Pattern Anal. Mach. Intell., 16(8):824-831, 1994.

[121] S. M. Khan, P. Yan, M. Shah. A Homographic Framework for the Fusion of Multi-view Silhouettes. In proceedings of ICCV 2007.

[122] B. Vijayakumar, D. Kriegman, and J. Ponce. Structure and motion of curved 3D objects from monocular silhouettes. In Proc. of CVPR96, pages 327334, June 1996.

[123] P. Favaro, S. Soatto. A variational approach to scene reconstruction and image segmentation from motion-blur cues. IEEE CVPR 2004.

[124] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan. Image Based Visual Hulls. In ACM Siggraph 2000.

[125] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, R. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. IEEE CVPR 2006.

[126] K. Wong and R. Cipolla. Structure and motion from silhouettes. IEEE ICCV 2001.

[127] Stauffer, C., Grimson, W.E.L. 1999. Adaptive background mixture models for real-time tracking, CVPR 1999.

[128] R. Szeliski. Rapid Octree Construction from Image Sequences. Computer Vision, Graphics and Image Processing, 58(1):23.32, 1993.

[129] S. Sullivan and J. Ponce. Automatic Model Construction, Pose Estimation, and Object Recognition from Photographs using Triangular Splines. IEEE TPAMI, 1998.

[130] A. Broadhurst, T. Drummond, and R. Cipolla. A probabilistic framework for the Space Carving algorithm. In ICCV, 2001.

[131] K. Kutulakos and S. Seitz. A Theory of Shape by Space Carving. IJCV, 38(3):199.218, 2000.

[132] O. Faugeras and B. Mourrain, On the Geometry and Algebra of the Point and Line Correspondences Between N Images, IEEE ICCV 1995.

[133] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: Motion deblurring via fluttered shutter. SIGGRAPH, 25(3):795 804, 2006.

[134] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. SIGGRAPH, 25(3):787 794, 2006.

[135] M. Ben-Ezra and S. K. Nayar. Motion-based motion deblurring. TPAMI, 26(6):689698, 2004.

[136] J. Jia. Single Image Motion Deblurring Using Transparency. In proceedings of CVPR 2007.

[137] P. Narayanan, P. Rander, and T. Kanade. Constructing virtual worlds using dense stereo. In ICCV, pp. 3.10, 1998.

[138] C. Hernandez and F. Schmitt. Silhouette and stereo fusion for 3D object modeling. CVIU, 96(3):367.392, 2004.

[139] S. Cho, Y. Matsushita and S. Lee. Removing Non-Uniform Motion Blur from Images.IEEE ICCV 2007.

[140] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *CVPR (1)*, pages 26–33, 2005.

[141] K. Mikolajczyk, C. Schmid. An affine invariant interest point detector. In: Proceedings of the European Conference on Computer Vision. (2002) 128142

[142] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

[143] H. Chang and D.-Y. Yeung. Graph laplacian kernels for object classification from a single example. In *CVPR (2)*, pages 2011–2016, 2006.

[144] J. Mundy, A. Zisserman. Geometric Invariance in Computer Vision. The MIT press (1992)

[145] M. Everingham, A. Zisserman, C. Williams, and L. van Gool. The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results. http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf.

[146] J. D. R. Farquhar, D. R. Hardoon, H. Meng, J. Shawe-Taylor, and S. Szedmák. Two view learning: SVM-2K, theory and practice. In *NIPS*, 2005.

[147] C. Wallraven, H.H. Bulthoff. Automatic acquisition of exemplar-based representations for recognition from image sequences. In: CVPR 2001 - Workshop on Models vs. Exemplars. (2001)

[148] L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *ICCV*, pages 1134–1141, 2003.

[149] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *ICCV*, pages 1816–1823, 2005.

[150] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *CVPR*, pages 443–461, 2005.

[151] A. Torralba, K. Murphy, and W.T. Freeman. Sharing Features: Efficient Boosting Procedures for Multiclass Object Detection, CVPR, vol. 2, pp.762-769 , 2004.

[152] J. Ng and S. Gong. Multi-view face detection and pose estimation using a composite support vector machine across the view sphere, Proc. Int.Workshop Recognition, Analysis and Tracking of Faces and Gestures in Real Time Systems,pp.14, 1999.

[153] T. Tuytelaars, L.V. Gool. Wide baseline stereo matching based on local, affinely invariant regions. In: British Machine Vision Conference. (2000) 412425

[154] S.Z. Li and Z. Zhang. FloatBoost Learning and Statistical Face Detection PAMI, 26(9):1112-1123, 2004.

[155] H. Murase, S. Nayar. Visual learning and recognition of 3-d objects from appearance. International Journal of Computer Vision 14 (1995) 524

[156] A. Pentland, B. Moghaddam, T. Starner. View-based and modular eigenspaces for face recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA (1994)

[157] Z.-G. Fan and B.-L. Lu. Fast Recognition of Multi-view Faces with Feature Selection ICCV, 2005.

[158] E. Bart, E. Byvatov, S. Ullman. View-invariant recognition using corresponding object fragments, ECCV, vol.2, pp. 152-165, 2004.

[159] V. Ferrari, T. Tuytelaars, and L. Van Gool. Integrating multiple model views for object recognition. In *CVPR*, volume 2, pages 105–112, 2004.

[160] S. M. Khan, P. Yan, and M. Shah. A homographic framework for the fusion of multi-view silhouettes. In *ICCV*, 2007.

[161] B. Leibe and B. Schiele. Scale-invariant object categorization using a scale-adaptive mean-shift search. In *DAGM*, pages 145–153, 2004.

[162] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, Nov. 2004.

[163] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *ECCV (4)*, pages 490–503, 2006.

[164] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *ICCV*, pages 503–510, 2005.

[165] E. B. Sudderth, A. B. Torralba, W. T. Freeman, and A. S. Willsky. Learning hierarchical models of scenes, objects, and parts. In *ICCV*, pages 1331–1338, 2005.

[166] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. Van Gool. Towards multi-view object class detection. In *CVPR*, volume 2, pages 1589–1596, 2006.