PEDESTRIAN DETECTION AND TRACKING USING STEREO VISION TECHNIQUES

by

Philip Kelly, B.A. (Mod)

Submitted in partial fulfilment of the requirements

for the Degree of Doctor of Philosophy

Dublin City University

School of Electronic Engineering

Supervisor: Dr. Noel E. O'Connor

December, 2007

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: _____
Philip Kelly (Candidate)

ID: _____

Date: _____

Acknowledgements

I wish to extend my heartfelt gratitude to the many people who have enabled and supported the work behind this thesis. Special thanks go to my supervisor Dr. Noel E. OConnor for valuable guidance and expertise as well as his constant support, advice and encouragement. Similar thanks are due to Dr. Eddie Cooke and Professor Alan Smeaton for their individual advice, support and suggestions at different stages of this work.

I would also like to thank Dr. Rachel McDonnell from the Graphics Vision and Visualisation group in Trinity College Dublin for her help in creation of the synthetic pedestrian dataset for disparity estimation evaluation. In addition, I would like to give special thanks Dr. Kieran Moran and his colleagues in DCU sports science for their expertise and access to their Vicon system. Also, thanks are due to both MERL (in particular Dr. Joe Marks and Dr. Paul Beardsley) and Dublin City County Council for their support of this project. Finally, this work would not have been possible without the financial support provided by Science Foundation Ireland via the Adaptive Information Cluster under Grant No. 03/IN.3/I361.

On a personal note, I would like to express my gratitude to my colleagues in the Centre for Digital Video Processing, for their debate, ideas, encouragement, expertise and various trips to various bars. To my parents and family, for unquestioningly supporting and encouraging me for as long as I can remember, in whatever I chose to do. Finally, to my friends for helping maintain my relative sanity, especially in the final months of this work.

Abstract

Automated pedestrian detection, counting and tracking has received significant attention from the computer vision community of late. Many of the person detection techniques described so far in the literature work well in controlled environments, such as laboratory settings with a small number of people. This allows various assumptions to be made that simplify this complex problem. The performance of these techniques, however, tends to deteriorate when presented with unconstrained environments where pedestrian appearances, numbers, orientations, movements, occlusions and lighting conditions violate these convenient assumptions. Recently, 3D stereo information has been proposed as a technique to overcome some of these issues and to guide pedestrian detection. This thesis presents such an approach, whereby after obtaining robust 3D information via a novel disparity estimation technique, pedestrian detection is performed via a 3D point clustering process within a region-growing framework. This clustering process avoids using hard thresholds by using bio-metrically inspired constraints and a number of plan view statistics. This pedestrian detection technique requires no external training and is able to robustly handle challenging real-world unconstrained environments from various camera positions and orientations. In addition, this thesis presents a continuous detect-and-track approach, with additional kinematic constraints and explicit occlusion analysis, to obtain robust temporal tracking of pedestrians over time. These approaches are experimentally validated using challenging datasets consisting of both synthetic data and real-world sequences gathered from a number of environments. In each case, the techniques are evaluated using both 2D and 3D groundtruth methodologies.

Table of Contents

List of Figures

List of Tables

List of Peer-Reviewed Publications

- P. Kelly, N.E. O'Connor and A.F. Smeaton. "Event Detection in Pedestrian Detection and Tracking Applications" in *The 2^{nd} international conference on Semantics and Digital Media Technologies (SAMT)*, Genova, Italy, 5-7 December, 2007.

- P. Kelly, N.E. O'Connor and A.F. Smeaton. "Pedestrian Detection in Uncontrolled Environments using Stereo and Biometric Information" in *The 4^{th} ACM International Workshop on Video Surveillance and Sensor Networks*, Santa Barbara, CA, USA, October 27, 2006.

- P. Kelly, E. Cooke, N.E. O'Connor and A.F. Smeaton. "Pedestrian Detection using Stereo and Biometric Information" in *The International Conference on Image Analysis and Recognition*, Povoa de Varzim, Portugal, 18-20 September 2006.

- P. Kelly, E. Cooke, N.E. O'Connor and A.F. Smeaton. "3D Image Analysis For Pedestrian Detection" in *The 7^{th} International Workshop on Image Analysis for Multimedia Interactive Services*, Incheon, Korea, 19-21 April 2006.

- P. Kelly, P. Beardsley, E. Cooke, N.E. O'Connor and A.F. Smeaton. "Detecting Shadows and Low-lying Objects in Indoor and Outdoor Scenes Using Homographies" in *The IEE International Conference on Visual Information Engineering, Convergence in Graphics and Vision*, Glasgow, U.K., 4-6 April 2005.

CHAPTER 1

Introduction

1.1 Motivation

Accurate detection and tracking of pedestrians are two essential components required by a variety of applications. The areas of application include, amongst others, Ambient Intelligence (AmI), automated surveillance, advanced user interfaces, image compression and content-based multimedia storage and retrieval [20]. Due to a large number of potential applications, pedestrian detection and tracking has become an extremely active area in computer vision research. The result of this has been a significant amount of prior art proposing pedestrian segmentation techniques using a myriad of approaches. Techniques include the application of traditional 2D computer vision techniques and the use of other sensor modalities such as infrared, laserscanners, sonar or radar. Many of the approaches proposed produce good results when presented with constrained scenarios that allow specific assumptions to be made. These constraining assumptions are generally introduced by techniques to reduce the number of complicating factors that are inherent in pedestrian detection thereby making the problem tractable. They include assumptions about the environmental conditions, the pedestrian appearance and flow density, the pedestrian and background colour intensity information, the length of time a person exists within the scene, that a person enters the scene un-occluded, and even the number of persons within the scene. Unfortunately, due to these assumptions, few approaches, if any, produce reliable results for long periods of time in unconstrained environments [21].

In this work, a robust pedestrian detection and tracking system is proposed that augments traditional 2D image processing with advanced stereo vision-based techniques. The proposed technique specifically targets relatively unconstrained environments and attempts to minimise such

constraining assumptions. In addition, it requires *no* external training[1] and yet is robustly able to handle:

1. occlusion, even when multiple people enter the scene in a crowd;
2. lack of variability in colour intensity between pedestrians and background;
3. rapidly changing and unconstrained illumination conditions;
4. pedestrians appearing for only a small number of frames;
5. relatively unconstrained pedestrian movement;
6. relatively unconstrained pedestrian pose, appearance and position with respect to the camera;
7. varying camera heights, rotations and orientations;
8. static pedestrians.

1.2 Application Areas

Pedestrian detection and tracking can be seen as a key enabling technology within the framework of a variety of intelligent systems. This technology is key to knowing who is where in a scene *and* what their actions have been. It potentially allows other layers in an application's framework to infer beliefs about those people. Accurate pedestrian detection and tracking is a prerequisite for a variety of computer vision based applications, such as automated security systems, and multi-disciplinary paradigms, for example Ambient Intelligence (AmI), to become viable. A high-level overview of some of these applications is now presented.

Ambient Intelligence (AmI) The vision of AmI [22] depicts environments that are able to adapt intelligently to facilitate the requirements of the people present. AmI leverages a networked system of smart devices and sensors, which have been smoothly integrated into the environment to act as a global interface between users and information systems [23]. In this way, the control of the augmented environment becomes action oriented, responding appropriately to the behaviour of the

[1] However, it is acknowledged that the designer has brought in his own area of expertise into setting a number of hard-coded thresholds throughout the system framework.

human users present. This promises many benefits for both single individuals and larger groups of people in a variety of application scenarios.

In order for AmI to become a reality, a number of key technologies are required from a variety of disciplines [22]. These include unobtrusive sensor hardware, wireless and fixed communication systems, software design, information fusion, intelligent agents, to cite but a few. In addition, the robust detection and tracking of humans in unconstrained scenes is a key enabling technology since knowing who is where in a scene *and* what their actions have been allows other layers in an AmI framework to infer beliefs about those people. Consider the example of an automated pedestrian traffic light system. An embedded intelligent system should be able to determine the number of people waiting to cross, whether any special assistance should be flagged for any individual pedestrian (e.g. wheelchair, children or elderly pedestrians), estimate the time needed for everyone to cross, determine the state of traffic flow on the road and ensure each person crosses the road successfully before allowing vehicular traffic to flow. Clearly detecting pedestrians is a necessary pre-processing step, but just because a person is in the scene doesn't mean that they want to cross the road. However, if the person walks towards the crossroads, stops and waits, then it can probably be assumed to be the case. Obtaining this information poses significant challenges when pedestrian detection and tracking in unconstrained real-world crowded environments are considered. RFID tagging is a possible solution for determining this in constrained environments, but cannot help in scenarios where there is no contact with people in a scene until they enter the environment.

Other example AmI applications include other public service applications, such as the counting and classification of pedestrians, for example the number of wheelchair users waiting at bus stops, taxi ranks, elevators, etc, so that the appropriate transport facilities can be dispatched if required. However, AmI envisions other advanced services in the home and office as well as in public spaces [22]. Such applications include automatically regulating the temperature and ventilation levels depending upon the number of people present in a room, smart meeting rooms that automatically summarise events, pedestrian collision warning systems within automated vehicles [24, 25, 26], and smart living rooms that will automatically pause a movie when a person leaves the room until he or she comes back [14].

Surveillance Another important application domain is that of surveillance. For the purposes of this discussion, a distinction is made between AmI and general surveillance applications. In this classification, the design of AmI applications is such the actions of people affect their environment

directly *and* the system reacts appropriately to the requirements of those present. These actions are generally apparent to the user and are designed to help or improve the standard of living of one or multiple end users. However, in the context of this thesis, surveillance applications are defined to be such that the actions of people within scenes do not affect the environment *or* the reaction of the system is such that it inhibits, rather than appeases, persons in the environment. For example, in some security applications the actions of a person can trigger security alarms. If this alarm results in the automatic locking of security doors, then the actions of that person has resulted in an affect on the environment. This is clearly a trait of AmI applications. However, unlike AmI applications, the changes to the environment are designed to impede not aid. This overlap between surveillance and AmI can be removed if the security system postpones such environmental changes for human verification of an event. As such, the application itself does not cause a change in the environment.

Surveillance applications can range from simply counting and tracking pedestrians for crowd density statistics and congestion analysis [27], to more sophisticated analysis leading to smart security surveillance systems, such as the one previously described. Statistical information of pedestrian numbers and flow has a variety of applications including the design and operation of facilities such as shopping centres, sports stadiums, business areas, airports and pedestrian road crossings [28]. This statistical information can also be used for commercial purposes, for example the management of shopping centres could charge rent for shops and advertising space in proportion to the pedestrian flow within the specific complex areas. In addition, statistical information is useful for security reasons, for example to know the occupancy counts of a building in case of a fire. These surveillance applications, however, require knowledge of only the pedestrian flow through a scene. More sophisticated surveillance analysis is generally required for security applications, for example detecting suspicious behaviour in public places, such as in airports and railways.

Content Retrieval Robust detection and tracking can also be considered in a content-based multimedia storage and retrieval scenario. For example, if typical CCTV surveillance video is augmented with accurate pedestrian tracking and statistical information it becomes possible to quickly search the video for specific events via the augmented meta-data. For example, if a lost child's appearance is known (for example the colour of their clothes and their height) then all possible detections of the child can be subsequently flagged from every camera in the system. Other examples include retrieving the number of pedestrians who entered a specific shop, or retrieving

footage of all persons who were in the vicinity of an area that was vandalised.

1.3 Challenges

The robust segmentation and tracking of pedestrians under unconstrained conditions introduces a multitude of complicating factors that make it one of the most challenging problems in computer vision. These complicating factors have to be acknowledged and addressed by computer vision systems if robust pedestrian detection is to become possible in real-world scenarios. This work strives to overcome as many of these challenges as possible.

1.3.1 Pedestrian Detection Challenges

For pedestrian detection techniques, the main challenges include;

Pedestrian Appearance A large variability in pedestrians' local and global appearance can be caused by various types and styles of clothing [29]. Therefore, basing a pedestrian detection technique on a particular feature, such as the appearance of legs, may not be applicable for all classes of pedestrian, such as those wearing a skirt.

Pedestrian Pose A pedestrian is a non-rigid body and as such a pedestrian's global shape can undergo a large range of transformations due to the variety of possible poses that can be assumed.

Pedestrian Orientation A pedestrian can be viewed at a variety of possible orientations with respect to the camera image. For example, the pedestrian might face the camera directly, i.e. at 0 degrees, or be front-parallel to the camera, i.e. at ± 90 degrees.

Pedestrian Position A pedestrian can be positioned in a scene at various distances to the camera. The appearance of pedestrians close to the camera can differ significantly from those at a greater distance.

Self Occlusion A pedestrian's silhouette may also be perturbed by a multitude of occluding accessories such as backpacks, hats, briefcases, and hand or shopping bags. In addition, various parts of a pedestrian's body, such as arms or legs, can be occluded by the orientation of the pedestrian to the camera.

Group Occlusion A pedestrian may be occluded by one or several other pedestrians, or objects, especially if the pedestrian is located within a crowd. A pedestrian can enter occluded and then remain occluded throughout the entire time-frame he/she appears in the scene.

Camera Field Of View A pedestrian may be only partly within the field of view of the camera. This may, or may not, indicate that the pedestrian is entering or exiting the scene. This can occur in multiple areas within the scene.

Non-pedestrian Objects Not all foreground objects in a given scene may be pedestrians, even if the shape and size are similar. It is difficult to predict and define models for all other real-world objects that may appear in a scene.

Environmental Conditions Pedestrian detection in real word scenarios brings extra difficulties including moving backgrounds, changing lighting and weather conditions, reflections on windows and shadows cast by pedestrians and other foreground objects.

Intensity Variation A lack of variability in colour intensity may exist between pedestrians, the background, and other foreground objects, resulting in difficult segmentation.

Time-frame A pedestrian may only appear in the scene for a limited number of frames, which can cause problems if a pedestrian detection technique is over-reliant on temporal data for classification.

In addition to these difficulties, techniques that have been developed for a particular camera height, rotation and orientation may not be applicable for a second camera position. For example, techniques in the literature that apply to an overhead, or birds-eye view, are generally not applicable when the camera is positioned at a more oblique angle.

1.3.2 Pedestrian Tracking Challenges

The issues presented thus far represent significant challenges for pedestrian detection techniques in single frames. However, the goal of many computer vision applications, such as those used for security, obtaining pedestrian flow information and smart rooms, require information not only about where people are at a given instant in time, but also *what* these people are doing. The information obtained at a discrete time instant is unlikely to provide such applications with this information. More useful information about the activities of people within the scene may be

obtained if the person is reliably tracked through time. Robust pedestrian tracking, however, introduces further challenges;

Pedestrian Movement A pedestrian may travel in a non-linear and unpredictable fashion, resulting in difficult tracking. For example, in real-world scenarios a given pedestrian may walk, run, stop or turn around unexpectedly.

Occlusions One or more tracks may merge into one. Depending on the tracking technique, when two or more tracked pedestrians occlude it is possible that one track can be temporarily lost. The tracking algorithm must be able to take this into account so that after occlusion, each track maintains the appropriate object (i.e. the one who was tracked before occlusion) [30].

Splitting A track may split into two or more pieces. This can be due to poor segmentation in the current frame, poor segmentation in the previous frames, or possibly due to a person depositing an object in the scene. If the reason is due to poor segmentation the system should be able to recognise this and remedy the situation.

New Tracks New pedestrians, which have not been previously tracked, must be recognised as new and not confused with previously tracked pedestrians.

Lost Tracks A track may be lost completely, due to noise, poor segmentation or the pedestrian leaving the field of view of the camera. This lost track should be marked as inactive and not confused with other tracks.

1.4 Objectives of this Thesis

The first objective of this thesis is to outline the current state of the art in pedestrian detection and tracking. This review outlines the most characteristic categories of techniques in both areas and briefly discusses the most interesting and promising techniques in each category. This thesis does not attempt to cover all aspects of pedestrian detection and tracking nor does it represent a detailed literature review of the vast amount of work published in these fields. Rather, it restricts itself to a discussion of the main techniques using both traditional 2D image processing and 3D stereo techniques.

The second, and most important objective, is to investigate a new approach to solving the problems related to pedestrian detection and tracking. One of the main goals is to explore the

robustness of such an approach under a variety of scenarios exhibiting a range of camera orientations, pedestrian flow densities, poses, and environmental conditions. In each scenario, each of the main processing modules incorporated within the proposed technique is rigorously evaluated.

The final objective of this thesis is to indicate directions for further research, namely to consider possibilities for further improvement of the proposed solutions and discuss the prospects of using them as a basis for a variety of end user applications.

1.5 Main Research Contributions

The proposed pedestrian detection and tracking system consists of three main analysis modules; (1) a disparity estimation technique that has been specifically developed for applications involving pedestrian detection; (2) a pedestrian detection module which, via an iterative region growing framework, clusters 3D information into pedestrian regions; and (3) a pedestrian tracking algorithm that incorporates a continuous detect-and-track methodology based on weighted bipartite graphs. Each module in the system can be viewed as a major contribution of the research programme documented in this thesis; however in addition, each module consists of a number of additional contributions.

The contributions in the first module include; (a) reducing the disparity search space via a 2D projective transformation of the input rectified images into *groundplane space*; (b) a novel technique to obtain highly reliable disparity matches, known as Ground Control Points (GCPs), to help guide final disparity estimation results; (c) the use of a *dynamic* disparity limit constraint in the disparity estimation process; and (d) a novel scanline cost function for the dynamic programming algorithm that enforces inter-scanline consistency in the final disparity map.

The second module includes; (a) a novel biometric person model; (b) a novel iterative region growing framework for person clustering; (c) the use of non-quantised plan-view statistics; and (d) a background subtraction framework that is highly robust to changing illumination conditions.

In the final module, contributions include; (a) a matching technique that incorporates a novel weighting scheme for matching pedestrians to previous tracks; (b) a series of kinematic constraints that model possible pedestrian movement through the scene and can be used to remove implausible matches of pedestrians to previous tracks; and (c) a post-processing stage to increase track robustness to occlusion and under-segmentation.

In addition to these research contributions in the context of image analysis, two further contri-

butions with regard to experimental evaluation are made. These contributions constitute techniques for evaluating the accuracy of stereo pedestrian detection algorithms with respect to both disparity estimation and pedestrian detection accuracy.

1.6 Thesis Overview

A review of the current literature on pedestrian detection and tracking approaches in both 2D and 3D are provided in chapters 2 and 3 respectively. In each chapter, the respective strengths and weaknesses of the various approaches are subjectively evaluated. In addition, in chapters 3 and 4 a review of the areas of multiple view geometry and stereo correspondence techniques are outlined respectively.

The remainder of chapter 4 and the next two chapters concern the three analysis modules of the proposed system that constitute the major contributions of this work. In each chapter, the proposed technique is outlined and evaluated using a dataset of challenging sequences taken from a variety of scenes. In chapter 4 an overview of disparity estimation techniques is given and the first contribution of this thesis is presented. In this chapter, a dynamic programming based stereo correspondence technique is presented that has been specifically developed for applications involving pedestrian detection. As discussed in chapter 5, this disparity map is then post-processed to remove artifacts and constrain the 3D points to a volume of interest, whereupon it is used as input to the pedestrian detection module. In this approach, the post-processed disparities are clustered together into pedestrian regions via an iterative region growing framework that incorporates a basic human biometric model which is automatically tailored for each pedestrian during their segmentation. The final module and contribution of this work is that of pedestrian tracking using a continuous detect-and-track methodology, which is described in chapter 6. This tracking framework involves the use of a weighted bipartite graph and a maximum-weighted maximum-cardinality matching scheme. In addition, kinematic constraints and explicit occlusion analysis are employed to obtain the best match from previous tracks to currently detected pedestrians.

In the final chapter, the thesis is summarised and a discussion on future work is presented that includes the potential use of the pedestrian detection and tracking system as the basis of a number of applications.

CHAPTER 2

2D Pedestrian Detection and Tracking

2.1 Introduction

In this chapter, an overview of 2D pedestrian detection and tracking approaches is presented and discussed in order to highlight their respective strengths and weaknesses. The chapter is broken up into three main areas; section 2.2 provides an overview of motion segmentation techniques which form the basis of many pedestrian detection techniques; a taxonomy of 2D pedestrian detection techniques is presented in section 2.3; and in section 2.4 a review of techniques applied for 2D pedestrian tracking is given.

It should be noted that in this chapter approaches such as [31], which attempt to track a group of pedestrians as a single entity, are not considered. In addition, techniques that are specific to alternative image sensing modalities, such as infrared [32, 33, 34] specifically considered. However, a number of techniques presented in this chapter have been applied to detect pedestrians in these modalities.

2.2 Motion Segmentation

Identifying moving objects is a fundamental and critical step in numerous pedestrian detection algorithms proposed in the literature. To this end, a variety of motion segmentation techniques are employed that aim to accurately detect and segment moving regions or objects from a temporal image sequence. For a given pedestrian detection technique, the choice of which motion segmentation technique to use depends upon both the specific requirements of the algorithm and a prior set of assumptions about the scenario, camera motion and environmental conditions. In

general three common motion analysis methodologies are applied by pedestrian detection algorithms; namely background subtraction [35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46], temporal differencing [47, 48, 49] and optical flow [31, 50].

Of these three approaches, traditional background subtraction, or suppression, is by far the most popular technique. However, some techniques, such as [31, 50] augment one approach – in their case optical flow – with another, such as background subtraction. In addition, temporal differencing can be thought of as a specific case of background subtraction based techniques, whereby the background model is set to correspond to a previous image frame.

2.2.1 Background Subtraction

In general, background subtraction approaches assume that there is a static camera and that image features, such as colour intensity or edge gradient information, of foreground objects differ to that of the background. In addition, an assumption is often made that there is little global change in illumination conditions, or that if a change does occur then it will be a small gradual illumination change.

Techniques of this type generally model the background with respect to relevant image features. Foreground pixels can then be determined if the corresponding features from an input image differ significantly from those of the background model. A variety of background models for use in pedestrian detection algorithms have been proposed. These models include; an image of the scene without any foreground objects [51]; the minimum, maximum, and largest inter-frame absolute difference for each pixel found during the training period [46]; the mean or the median value of a pixel in the previous number of frames [52, 50, 53, 54]; modelling the variance of a pixel by a uni-modal distribution, like a Gaussian distribution [55, 56, 57, 58, 4]; modelling the variance of a pixel by a Mixture of Gaussians (MoG) [59, 60, 61, 15, 62, 63, 64]; using non-parametric models [65]; and using codebook-based background models [66, 67, 68].

Many of these models, such as [69, 1, 70], are applied to traditional colour intensity values. However, other techniques model background image *features* rather than intensity values, due to the robustness of certain image features to changing lighting conditions. Such feature based background models include that of [70] which models image gradients and [71, 72] which maintains background models of edge features. These background models, however, tend to result in sparse background models meaning that dense foreground maps are not obtained. Recently, a number of pedestrian detection techniques, such as [5, 73, 74, 75], have begun to maintain background

models based on range information. This is due to range information being a powerful cue for foreground-background segmentation [76]. In addition, compared with the intensity-based approaches, range-based foreground segmentation is less affected by lighting conditions including shadows. However, range based models do have a number of drawbacks. They cannot correctly segment objects that are at the same depth as background regions, for example, a person standing beside a wall may not be segmented correctly. In addition, range information based on disparity estimation techniques can be a noisy modality where the standard deviation of the depth value at a pixel over time is commonly of the order of 10% of the mean [21]. This can result in a noisy segmentation, especially within regions of homogeneous colour. In order to overcome some of these issues, some background models [77, 14] employ disparity estimation techniques which only obtain disparity within regions of high texture. However, this technique results in sparse disparity maps, which can be problematic as the model may contain regions where the background is invalid. Finally, some pedestrian detection techniques, such as [14], apply more than one background model. For example, in [14] a range model is augmented with a traditional colour background model.

The majority of the models applied in pedestrian detection and tracking techniques in the literature are based on colour intensity values. However, robust background-foreground segmentation from these background models is not a trivial problem. A number of significant and difficult challenges exist which are caused by moving backgrounds, weather conditions, shadows and static foreground regions [78]. In addition, significant problems can be introduced in unconstrained environments due to global illumination changes, such as that presented within the image sequence of figure 2.1[1], where a large change in illumination occurs around multiple occluding pedestrians over a short number of frames. In order to cope with such illumination changes, many techniques incorporate a background model update parameter into their algorithmic framework. However, in order to cope with sudden illumination changes, such as that of figure 2.1 the update parameter must be set so that the background model is updated quickly. This can cause problems in other areas, for example if the background is updated too fast, then people who stop momentarily can get absorbed into the background. For background subtraction techniques, a number of similar trade-offs exist whereby an increase in accuracy in one area can lead to a reduction in overall robustness in a second complementary issue.

To date there is no background subtraction technique that addresses all the traits required

[1]These images form part of the test dataset used for evaluation of the proposed algorithms within this thesis.

of background models in unconstrained environments. The reasons for these failings lie in the fact that, in real-world conditions, the background subtraction technique is, at best, attempting to solve a two-class classification problem, where one class is undefined [79]. The background can be modelled by observing a pixel over time and making assumptions, such as constant lighting conditions. However, the alternative category, *not-background*, cannot be modelled sufficiently well *a priori*. Therefore, if an object similar to the background model is introduced, or if the background model changes unexpectedly, errors are inevitable. However, motion segmentation techniques remain the basis of many pedestrian detection and tracking techniques, whereby only pixels declared as foreground are processed further to obtain hypotheses of pedestrian objects. As a result, techniques that are built upon this basis are limited by the success of the underlying flawed segmentation algorithms of motion segmentation. It can be argued, therefore, that if background subtraction techniques are to be applied within pedestrian detection and tracking algorithms then it may be more fruitful to use the resultant foreground regions solely to *guide* the final segmentation of objects as opposed to being the basis of a technique to obtain those objects. This is the methodology employed in the proposed pedestrian detection system in this thesis, where two background models are employed; one based on range information and the other on edge gradients.

2.2.2 Optical Flow

The application of optical flow techniques as an alternative to background subtraction or temporal differencing approaches are adopted by some pedestrian detection algorithms to segment moving foreground regions. Optical flow determines the 2D projection of the 3D velocity map of the scene on the camera plane [31]. As a result, all the movements along the cameras optical axis (i.e. the cameras z-axis), which are not perceived by the sensor, can not be computed by optical flow [31]. Optical flow can therefore be thought of as a velocity field in the image which transforms one image into the next image in a sequence [80].

Motion segmentation based on optical flow has some advantages to segmentation based on background subtraction techniques. For example, discontinuities in the optical flow can help in segmenting images into regions that correspond to different objects [81]. This can be used to a great advantage when segmenting rigid regions, such as cars, as the whole of the object has similar optical flow. The segmentation of non-rigid regions, such as pedestrians, is however non-trivial as the optical flows of regions described, for example, by the left and right arms and legs of a walking person are different [82]. Therefore, segmentation based on optical flow techniques is

Figure 2.1: Shadows: change in illumination over 14 frames.

difficult to apply within pedestrian detection algorithms due to the non-rigid motion of pedestrians [3]. In addition, optical flow based segmentation techniques also suffer from other disadvantages, including that they cannot be used to detect static pedestrians within a scene.

In addition to these problems, the actual computation of accurate optical flow is non-trivial, computationally complex, sensitive to noise and cannot be applied to video streams in real-time without specialised hardware [83]. Optical flow computation algorithms also suffer from many of the problems that disparity estimation techniques endure, see section 4.2.1, where specular effects, shadows, insufficient texture (the aperture problem) and occlusion make the unambiguous recovery of the true motion field impossible [80]. In addition, the non-rigid movement of pedestrians can cause noisy results, as optical flow algorithms tend to fail in regions where there are multiple motions, occlusion, and non-rigidly moving areas [39].

To reduce some of these problems, pedestrian detection techniques tend to augment optical flow with other assumptions. For example, [84], applies an optical flow-based person tracking method using multiple cameras in indoor environments, but assumes that there is only one person in the image at a given time. It therefore tracks the region in the image with the highest uniform flow vectors. Other techniques, such as [31] and [50], initially apply background subtraction techniques to obtain foreground regions. Optical flow is then computed on foreground regions only. Finally, [50] uses optical flow to remove false positives obtained from background suppression, noting that *ghosts*[2] can be removed as they have a near-to-zero optical flow [50]. However, for reasons outlaid in this section, a motion segmentation technique based on optical flow is not adopted in the proposed pedestrian detection and tracking technique.

[2]Ghosts are introduced when objects, which are incorporated into the background model, move. When this occurs two new foreground regions are detected; the region where the object has moved to, and also the region where the object was previously located in the background model. The second region is referred to as a ghost, since it does not correspond to any real moving object [52].

2.3 2D Pedestrian Detection

Many pedestrian detection techniques have been proposed in the literature. Some detect pedestrians using temporal data obtained from a sequence of frames, whereas other approaches attempt to solve the harder problem of recognising pedestrians in single images. As discussed in section 1.3.1, there are many challenges facing such techniques such as occlusions, environmental conditions, lack of intensity variations between objects and the huge amount of possible variations in the orientation, pose, size and appearance of humans in a given scene.

To cope with these problems, many techniques simply ignore some of these possibilities. For example, assumptions are made that; pedestrians do not occlude [85]; all moving objects in the scene are pedestrians [28]; a pedestrian is wearing tight-fitting clothing [82]; a pedestrian's feet are always visible [39]; a pedestrian is fronto-parallel to the camera image plane [29]; skin colour pixels [86] or faces [73] or heads [35] are visible; people enter the scene one at a time [57]; or pedestrians enter the scene in predefined areas [69]. Other techniques, such as [79, 87], make an assumption that a person appears in the scene un-occluded for a given period of time allowing a model of the pedestrian to be built up while they are isolated. These, however, are assumptions that should be avoided within the framework of the proposed approach.

In this section, pedestrian detection approaches in single frames based on 2D techniques are presented. For the review, the techniques are split into seven categories;

- *Foreground Blobs*, whereby blobs are obtained from background subtraction and examined using analytical techniques;
- *Template Matching*, where a single, or multiple, 2D model(s) of a pedestrian or parts of a pedestrian are used;
- *Explicit 3D Shape Models*, whereby a single 3D model is used to describe a range of possible pedestrian poses and orientations;
- *Statistical Shape Models*, which use a single 2D model to describe the main modes of variation in a pedestrian's appearance;
- *Low Level Features*, whereby low-level features are extracted and pattern classification techniques are applied to determine the presence of a pedestrian;

- *Multi-Cue Based Approaches*, where multiple cues, such as face and skin detection are used to increase reliability;
- *Multi-Frame Approaches*, whereby temporal information, gained over a number of frames, to classify a foreground object as being a pedestrian.

It should be noted that any given approach defined in the literature may cover more than one of these categories and as such may be cited in more than one of the following sub-sections.

2.3.1 Foreground Blobs

The first generation of person detectors, such as [28, 85], make several assumptions regarding the nature of the objects in the scene, including that after background subtraction each foreground blob represents a separate pedestrian. These assumptions lead to them being prone to error in unconstrained conditions, when due to occlusions and background subtraction errors, this person-to-blob relationship fails. These assumptions were relaxed in other techniques, such as [88, 89, 44, 46], where blobs are associated with one or more people via tracking information from previous frames. In these approaches, if a foreground region is detected that does not sufficiently overlap any of the existing objects, it is assumed to belong to a single, new and un-tracked person. It is clear, however, that these techniques are not applicable to the real-world scenarios that this thesis wishes to address, where multiple pedestrians may enter the scene simultaneously and occluded.

2.3.1.1 Overhead View

A major problem with these basic techniques is that individually pedestrians are assumed to be segmented in separate blobs after background subtraction. This means that there is an inherent constraint in the system that states that occlusion should never occur. This is of course untrue in real-world scenarios. A technique to reduce occlusions, applied in some papers [69, 90, 91, 71, 92, 42], is to mount the viewing camera overhead and point it toward the ground.

In [69, 93, 91] background subtraction techniques are applied to segment objects from the background, and the foreground pixels are clustered into 2D blob regions using connected components. These regions are less likely to include more than one pedestrian when compared to more oblique camera views. In addition, the overhead viewpoint has additional advantages, including, due to the camera positioning, the fact that the size of pedestrians is relatively consistent. This can be exploited for tracking and counting purposes. Using this information, even if two or more

people are segmented into a single blob, using the knowledge of the average area in pixels of a person from a given top view, it can be easily determined how many people form a given region [42].

This viewpoint, however, does have disadvantages. Firstly, the camera orientation is generally only applicable to indoor scenarios due to the necessary overhead camera placement structures being unavailable in outdoor environments. Secondly, a camera in this point of view generally has a limited field of view [15], as a maximum height is constrained by a ceiling. To counter this reduction in height, techniques generally employ wide-angle lenses to increase the field of view [42]. With the use of these wide angle lenses, however, significant occlusion problems can be encountered in all but the central area of the image [15]. Therefore, unless the surveillance area is small and indoors, this viewpoint does not generally have many advantages over other camera setups.

2.3.1.2 Blob classification

From more oblique camera viewpoints, more sophisticated pedestrian detection algorithms exist that incorporate techniques to discriminate blobs caused by pedestrians from other objects. For example, in [35] pedestrians are disambiguated from other objects by applying a simple 2D stick figure model and ensuring that a foreground blob consists of a head region and a rectangular torso region that are of specific ratio of about 4 to 1. Any region that is consistent with the coarse human model is considered a human subject. Other simple metrics are applied in [47, 94, 95] whereby blobs are classified as belonging to either human or vehicular objects. However in these cases, if a region is classified as being a group of people, then the authors do not attempt to segment each individual person.

A similar technique is applied in [96, 97]. In this approach, a difference image, d, is first obtained using background subtraction. A projection, P, is then created across the X-axis of d, where $P(x) = \frac{1}{n}\sum_{y=1}^{n} d(x, y)$, where n is the number of image rows. P can be thought of as a 1D histogram. The width of this histogram is user defined and usually it is less than or equal to the width of the input image. An assumption made in this technique is that if there are two or more people in a foreground blob, then the distance between them will be such that there is be a significant rise and fall in the vertical projection histogram between the two people. It therefore subjects P to a fixed threshold. $P(x)$ is considered part of a valid object only if it is greater than this threshold, all other foreground pixels are discarded. After this process, the foreground is re-

clustered using a connected components algorithm [12]. A similar approach is applied in [56], where a person in the vertical projection histogram corresponds to a distinct peak region, which is greater than a threshold, occurring between two major valleys, which are lower than a second threshold. These techniques are an improvement from the previous vertical projection techniques as it allows pedestrians to be within closer proximity within a given scene.

Vertical projection approaches make use of a simple geometric human model, which assumes that a human body forms a peak in a projection of foreground pixels onto the X-axis. Human shapes are extracted by searching the projection for relevant peaks and troughs. Detecting human shapes using this technique allows significant variability to occur in the human shape. This can be both an advantage, in that it detects humans of multiple shapes and orientations, and disadvantageous, as other objects can occur in the scene that displays the same properties. In addition, these techniques do not detect two people when they get significantly close or when one person moves behind a second causing occlusion, nor do they give an indication of how good a match is but instead return a binary value of 1 or 0 indicating that a person has been detected or not.

2.3.2 Template Matching

Template matching can be used to overcome some of the problems within vertical projection techniques. The idea is to create a model of an object of interest (called the template, or kernel) and then to search within an image, or foreground blobs, for objects that match the required template. In general, for a given template position, scale and orientation, a similarity measure is obtained that represents the quality of match between the features of a template and the query image. If the similarity measure is above a threshold, then a possible matching of the template is reported. The main issue with template matching is obtaining a template (or set of templates) that are robust enough to detect the wide variety of poses that a pedestrian may exhibit. Approaches proposed to achieve this tend to either use generalised templates that encompass only the main features of the human body [5] or multiple templates of more detailed person silhouettes [98].

If the first technique is employed, then the template must be robust enough to detect pedestrians in multiple poses, but not over generalised so that many false-positives occur. Templates of this sort applied in pedestrian detection techniques include; a circular curve to detect people from overhead views [71]; binary templates for detecting head regions from foreground infrared blobs, or edges and colour information, see figures 2.2(a) and (b) respectively; an Ω-shaped head and shoulder contour template for use with edge features [3], as shown in figure 2.2(c); a similar

(a) (b) (c) (d) (e)

Figure 2.2: Templates; (a) Infrared head template [1]; (b) Eight head templates [2]; (c) Head and shoulder template [3]; (d) Head and shoulder template with normals [4]; (e) Body template at various scales [5].

Ω-shaped template, applied in [4], which also computes the normals of the contour points that are incorporated into the matching framework, see figure 2.2(d) where the red lines represent the normals; and person templates that encode both head and torso shape for stereo disparity matching [5], as illustrated in figure 2.2(e).

These templates, however, are fairly generic and exploit only general properties of the human body. As such they may result in a high false alarm rate, especially in areas of rich texture where there are abundant edges of various orientations. However, the use of more explicit object models, as used in [6], can significantly improve reliability [99]. In [6] an individual template – such as that of figure 2.3(a) – is created for each particular pedestrian pose and orientation within the entire pedestrian state space. In [6] the relevant templates are obtained from a database of 1100 pedestrian silhouettes, normalised for scale. However, this increase in template numbers can introduce problems with respect to the computational expense in matching, especially if each template has to be searched at different positions, orientations and scales. This number of possible combinations grows exponentially with increasing template numbers. It may therefore become intractable to search for the best match with respect to each of the templates [100].

Different approaches are taken to reduce the computational expense. Most systems assume pedestrians are standing upright, so only one template orientation is searched for; others use stereo depth information of foreground objects to determine the relevant scale size [5]; or apply a coarse-to-fine approach [33] whereby a scaled template match is performed at a low resolution with a relatively low threshold, for successful matches the resolution is increased and matching is reiterated with a higher threshold; other techniques [95] only perform the search in certain *detection areas* within the frame. Finally, in [6] an attempt to reduce this expense is made by using a coarse-to-fine approach over a template hierarchy, see figure 2.3(b), whereby matching involves a depth-first traversal of the template tree structure, see figure 2.3(b). This hierarchical setup allows pruning of the search space which brings large increases in efficiency to techniques employing multiple templates. However, the setup of the hierarchy can be problematic. In [6] a database

(a) (b)

Figure 2.3: Template hierarchy [6]; (a) Individual template; (b) Template hierarchy.

of 1100 pedestrian shapes using 5 scales leads to 5500 shape templates. Similar approaches are implemented in [29], using a set of 210 pedestrian silhouettes (plus their mirrored versions) and 7 scales, and in [101], which limits template shapes to the head and upper body.

The performance of a template hierarchy approach is hampered by the fact that the matching remains dependent on a reasonable contour segmentation [98]. In addition, as the numbers of templates increase, so too does the false alarm rate as clutter and poor contrast can cause the highest-ranking template match to lie on non-pedestrian objects [29]. To reduce the false alarm rate, the template hierarchy approach is usually supplemented with a post-processing step such as colour background subtraction [101], a second complementary pedestrian detection approach [29], or stereo information. Finally, it is noted that although the template hierarchy can capture the variety of object shapes, it cannot handle large shape variations appropriately when pedestrians are very close to the camera [102], nor can it perform well for camera setups that differ significantly from those used to create the search templates. For these reasons, the use of 2D templates is not applied in the proposed pedestrian detection system in this thesis.

2.3.3 Explicit 3D Shape Models

All the pedestrian models discussed thus far are 2D models representing the shape of a pedestrian that has been projected onto a camera image plane. Using these techniques, either a simple geometric human model or a number of more explicit models are required to cover the entire state space of pedestrian poses and orientations. However, issues can arise if a pedestrian's pose differs significantly from all those represented in the 2D models. In order to address this issue, and to simplify the creation of the required pedestrian models, some techniques [8, 7, 102] replace these multiple 2D models with a single *3D* model, which incorporates a certain number of degrees of freedom. In general, these techniques then adopt a *hypothesis-and-verify approach* to detect pedestrians in a particular image. Initially a particular number of pedestrians are hypothesised

with respect to their pose, orientation and position. These person parameters are then synthesised using the 3D model(s). Each model is then projected onto the image plane and compared to the input image using techniques similar to template matching. Finally, optimal 3D models can be obtained using search techniques.

The advantage of this approach is that, if a correct match is found, then the pedestrian location, orientation *and* pose can be determined. This information can be valuable for tracking and some end-user applications, such as virtual gaming systems and smart rooms. However, an issue with these types of shape models is that, as the human body shape is highly articulated, a large number of degrees of freedom are needed to adequately cover the required number of pedestrian orientations and poses. In [7, 103] a 3D model of the human head, arms and torso with 14 degrees of freedom (DOF) is used; 3 DOF are used for the positioning of the root of the articulated structure; 3 for the torso; and 4 for each arm – see figure 2.4(a). However, it is stated in [4] that to model the human body precisely, a kinematics model with over 20 degrees of freedom is required. A model consisting of 20 DOF is employed in [8], which is then extended to 22 DOF in [104] – see figure 2.4(b). The problem with increasing the number of DOFs is that it leads to an explosive increase in the search space, thereby making it extremely computationally expensive to search for the best solution within each DOF. Therefore, many of these techniques make assumptions about the pedestrians in the scene, such as in [8] where it assumed that the initial pose of the person is known. This is achieved by manually setting the model in the first frame of a sequence. The 3D model is then used to track the person in the remainder of the sequences.

In [4], however, the search space is decreased by employing a simple 3D ellipsoid model for the human shape, whereby the authors do not attempt to capture the detailed parameters of the human body. Instead, the human shape model is approximated by four ellipsoids corresponding to head, torso and two legs, where each ellipsoid is controlled by two parameters called length and fatness. However, the authors consider only three articulations; both legs together, for a person who is standing still; left leg forward and right leg forward for a walking person. The standing model has two orientations, from the front, 0^{o}, and the side, 90^{o}. Each walking model has six orientations, namely $\{0^{\mathrm{o}}, \pm 30^{\mathrm{o}}, \pm 60^{\mathrm{o}}, 90^{\mathrm{o}}\}$. There is therefore only one parameter needed for the model and orientation, which could be one of 14 values – see figure 2.4(c) for all 14 models and orientations. The parameters for each human object is therefore $\{l, x, y, h, f\}$, where l is the model and orientation, (x, y) is the position of the model, h is the height and f is the fatness. The solution to the model-based segmentation problem, which is obtained using an efficient Markov

(a) (b) (c)

Figure 2.4: Explicit 3D models; (a) 3D model [7]; (b) Full 3D body model [8]; (c) 14 3D models [4].

Chain Monte Carlo (MCMC) technique, includes the number of objects in the scene and their associated parameters. This technique has illustrated good results for tracking multiple people with a single camera slanting at about 45 degrees from 5 to 15 metres from the subjects. However, if the visual contrast between occluding and occluded persons is low, it often fails to track the occluded person [105]. In addition, the technique requires good foreground segmentation and edge detection in order to extract the required landmark points for heads, which can be challenging in unconstrained environments [106]. Finally, as with 2D template matching techniques, the approach cannot handle large shape variations appropriately when pedestrians are very close to the camera.

2.3.4 Statistical Shape Models

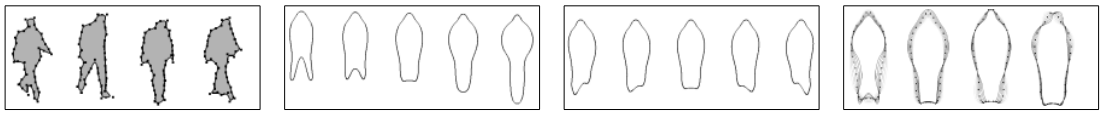
For pedestrian detection techniques the biggest challenge lies in the significant amount of variations in the shapes, pose, size and appearance of humans [33]. To cover this pedestrian state space the hierarchical template matching approach of [98] made use of 1100 explicit templates obtained from pedestrian silhouettes. Alternatively explicit 3D models with over 20 degrees of freedom are required [4]. An issue that affects both these approaches is that even when they are presented with a possible pedestrian region it can become extremely computationally expensive to verify or reject the region as being a pedestrian based on these models. This is primarily due to the scale of the search space that needs to be traversed.

However a third approach, based on Point Distribution Models (PDMs) has also been applied for pedestrian detection and tracking techniques [9, 99, 107, 53]. In these approaches, 2D statistical shape modeling techniques are employed to incorporate the wide variability of pedestrian shapes into a single 2D model. The PDMs generate their statistical description of the shape and variation of the models off-line from a set of training input data – see figure 2.5(a). Each shape in the training set is represented by a set of n labelled landmark points. These training vectors

in their current form result in an n dimensional space, which as before, can result in a very large space to search for similar shapes. Therefore, after the creation of this nD search space, a dimensional compression of the representation is achieved using Principal Components Analysis (PCA) whereby the m main modes of variation in the training data are determined and kept, but the lesser modes of variation and those due to noise are removed. Figure 2.5(b) shows the first mode of variation within the PDM of [9]. The second and other modes of variation are presented in figures 2.5(c) and (d) respectively. In general, the choice of m should depend on the variation seen in the training set and then altered to account for a user defined percentage of this variation. For example, in [108] the authors believe that fewer than 10 modes are required to account for most of the variability in their pedestrian data. Alternatively in [9], which used 40 control points on its training data, the first 18 models of variation are used as they account for 90% of the variance present in the training data. However, [109], which models the Ω-shape of a person's head and shoulders with 23 control points, uses only the first 5 modes of variations within its shape space.

The resultant models have similar properties to that of explicit 3D models, in that they allow a variety of plausible shapes to be fitted to new data by varying parameters. This is achieved by varying the m shape parameters within limits learnt from the training set. However, the statistical shape model has the added advantage of being able to verify or reject an initial hypothesis in an efficient manner. Given a new foreground image, in which the location of an instance of a modelled shape is required, the parameters for each mode of variation in m and the translation, rotation and scale of the model must be determined. However, using a *hypothesis-and-verify approach*, as was implemented using 3D models, can be result in slow convergence [12]. A quicker approach is to use the PDM as the basis of an Active Shape Model (ASM), which can be thought of as an iterative framework for the fitting of a PDM. In this process, the shape is initialised using an initial prediction that results in an approximate fit. The model is then iterated toward the best fit by locating improved positions using landmark points. After this the PDM parameters are recalculated and the technique is reiterated until it converges. This technique is applied in [53, 110, 99, 109].

The major drawback of using ASMs, apart from the usual problems associated with edge extraction and background subtraction techniques, is obtaining a good initial hypothesis for segmentation, especially in crowded environments. Head positions determined from template matching based techniques are used in [53, 109] as an initial estimate. An ASM model is then fit and used to filter out false-positives from the template matching based approach. In [99] two hypotheses



(a) (b) (c) (d)

Figure 2.5: PDM from [9]; (a) Training examples; (b) First mode of variation; (c) Second mode of variation; (d) Other modes of variation.

are used; that the top of the region bounding box corresponds to the top of a person's head, and that the bottom of the region bounding box corresponds to the bottom of a person's feet. A second issue associated with ASMs is that the training of the PDM can be problematic. The n labelled landmark points should be consistent from one shape to the next and this alignment of training data is not trivial, especially in the presence of occlusion where all landmark points may be visible in each training example. A more common approach to obtain these feature points is to use background subtraction techniques to segment the pedestrian's silhouette, which is then approximated with a uniform β-spline where the control points are placed at approximately uniformly spaced intervals along the contour [107]. However, using this technique not all landmark points may correspond exactly to the same feature points on the silhouette. To cope with these problems [107] uses a weighted least squares method to align two shapes, where the weights are chosen so that more significance is given to the more stable landmark points, i.e. the points which vary their position the least over the entire training set.

Finally, it is noted that ASMs share many of the issues involved with pedestrian detection from templates, including its inability to handle large shape variations when pedestrians are very close to the camera or detect pedestrians from varying camera setups. Therefore, as with templates, the use of ASMs (or PDMs) is not applied by the author in the proposed pedestrian detection system.

2.3.5 Low-level Features

Another line of approach, which shares some similarities with template matching, involves shifting windows of various sizes over the image at different resolutions, extracting low-level features, and using pattern classification techniques to determine the presence of a pedestrian [33]. As with statistical shape models, techniques of this type are trained from an off-line pedestrian dataset where multiple sets of windows at various resolutions are examined. Features, typically designed to be illumination change invariant, are then extracted and aggregated across these windows. The best features that define the pedestrian class are chosen and used to train a classifier, such as a Support Vector Machine (SVM) [111, 112, 10, 113, 11, 34, 114]. These techniques can generally

be applied without the application of motion segmentation techniques. However, some techniques do employ background subtraction as a post-processing step to remove false-positives. A number of such techniques have been proposed in the literature using a variety of features including Haar wavelets [112, 111, 115], normalised Histogram of Oriented Gradient (HOG) descriptors [114, 10] and edge density magnitude features [116].

In [112] once training of the classifier is completed, the system is able to detect pedestrians at arbitrary positions by shifting the 128×64 window across the input image, thereby scanning all possible locations. At each location, 29 features, which were manually chosen to be the features of the pedestrian class, are extracted from the window. These 29 features were then used to train the SVM with both pedestrian and non-pedestrian images. However, when using this technique to search an input image for pedestrians, an extremely computationally expensive search procedure must be undertaken. In addition, the results of this are likely to cause multiple responses from a single pedestrian [102].

The overall accuracy of a given learning algorithm can be improved by applying a general method known as *boosting* [117]. This is usually achieved by implementing the Adaptive Boosting (AdaBoost) algorithm [118]. The principle of the approach is based on the belief that a highly accurate or *strong* classifier can be produced through the combination of many inaccurate or *weak* classifiers [10]. The main idea of AdaBoost is to assign each example of the training set a weight in an iterative framework. At the beginning the weights of each training set example are equal. The algorithm then selects the weak classifier that minimises the error with respect to the classification of the training examples. For this weak classifier, the weights for each training example that are *incorrectly* classified by a weak classifier are increased. Therefore, when the algorithm is testing the remaining weak classifiers, it will select a classifier that better identifies those examples that the previous weak classifier missed. This leads to an increased focus of the algorithm on the more difficult examples of the training set. This technique is reiterated, where the next best weak classifier is selected that, *in addition* to the previously selected and weighted weak classifiers, minimises the error with respect to the classification of the training examples. This iteration is stopped when either all the weak classifiers are used, or the error of the best remaining weak classifier is less than a threshold. In this approach, the weights of each weak classifier are determined by a weighted majority vote, where training examples with lower classification error have higher weight. Therefore, using this technique AdaBoost can be used to both to select the set of features and train the classifier [119]. AdaBoost was used to train a number of classifiers used in

pedestrian detection algorithms, including those applied in [10, 115, 120, 117].

In addition to improvements in accuracy by training using AdaBoost, the overall complexity of these approaches can be reduced by organising the weak classifiers into a collection of cascaded layers [117]. The overall form of the detection process is that of multiple layers of classifiers of increasing complexity. A positive result from a simple first classifier triggers the evaluation of a second, more complex, classifier, where a second positive result triggers a third classifier, and so on. A negative outcome at any point leads to the immediate rejection of the sub-window. Therefore, the simpler classifiers are used to reject the majority of sub-windows before more complex classifiers are called upon to achieve low false-positive rates [119]. This technique is applied in [115] whereby a classifier cascade is created for pedestrian detection and AdaBoost is used to train the cascade nodes. However, the system was only trained on full human figures at low resolution and does not detect occluded or partial human figures.

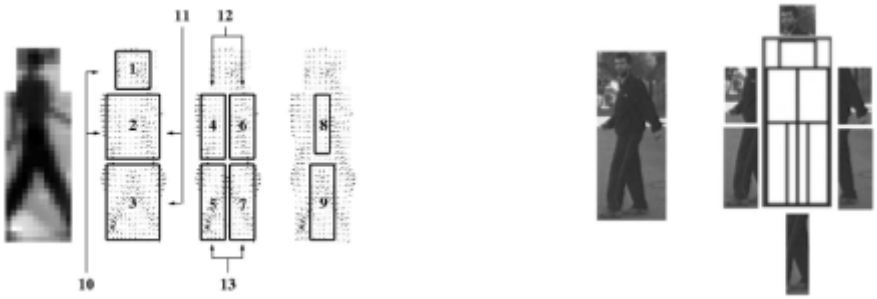
A similar approach is applied in [10], the crucial difference being that the search window is broken into 9 sub-windows and trained using HOG descriptors – see figure 2.6(a). Each sub-window, and the four pair combinations – as seen in figure 2.6(a) – is then broken into cells and processed. A separate SVM is then used to classify each local region independently. The results of each separate classifier are then fed into a final SVM that classifies the window as being a pedestrian or not. The idea behind the division of the window is that there is believed to be too much variability in a pedestrian pose for a single SVM to classify. The sub-windows are therefore used to breakdown the overall variability of the class into manageable pieces that can be captured by relatively simple component classifiers. A similar approach using 6 subregions is undertaken in [113, 11] – see figure 2.6(b) – however 7 different features are examined for each region, including Haar wavelets features, gradient based features, intensity features and texture features. The best performing features for each region are then employed for the hierarchical SVM classification.

These techniques all suffer from a variety of problems that include the inability to detect persons of varying scale, especially when pedestrians are close to the camera. In addition, as the SVM has been trained to detect a pedestrian centred in the window, mis-classification can occur if their position is not exactly centred [112]. Finally, the training of SVM classifiers can be problematic. The ratio between positive and negative samples has to be set to an appropriate value. A very large number of positive samples in the training set may lead to a high percentage of false-positive detections during on-line classification. On the contrary, a very large number of negative samples produces mis-learning [11]. In addition to this, the size of the training dataset must be

large enough to represent the problem well and the quality of both positive and negative samples can have a large effect on the classification results from test data. The negative samples should account for ambiguous objects, such as poles, bicycles, etc, while the positive examples should well represent the various pedestrian poses, orientations and appearances. Finally, the content of the test set must have similar characteristics to those of the training sets in terms of illumination conditions, variability, ratio of positive/negative samples and quality of negative samples [11].

However, not every technique in this class employs the use of SVMs. One such example is that of [29]. In this approach, evidence is integrated in multiple iterations and from different sources. This information is then used to extract pedestrians via a Generalised Hough Transform based approach. During training, image patches are extracted around features extracted using a scale-invariant interest point detector. A high quality foreground segmentation mask is associated with each image patch, including the relative position to its object centre. Clusters of similar image patches are created. Starting with each patch as a separate cluster, the two most similar clusters are merged as long as the average similarity stays above a certain threshold t. Note that using this technique, two image patches in a particular cluster may come from two different training images, or two different parts of the same image. For a test image, image patches are again extracted with the same interest point detector, and the patches are again matched to the codebook using the same threshold t. As previously noted, as each codebook match can have several valid interpretations, each image patch in the codebook entry casts votes for possible positions of the object centre. Hypotheses are then obtained as maxima in the continuous Hough space. This technique allows multiple scales of objects to be detected but has only been shown to apply on people walking parallel to the camera image plane. The main restriction of this technique is that, as it is based only on local features, it has a very limited notion of global consistency [29]. This lack of global knowledge can cause, for example, three legs to be added to a single hypothesis. This may augment its recognition score, leading to important evidence being withheld from neighbouring hypotheses and thus to lost detections. To enforce this consistency, template matching, is employed to to verify and refine object hypotheses, whereby the best hypothesis is determined via the overlap of template matching scores and original hypothesis.

Due to the issues inherent in these techniques, especially the problems with respect to detecting pedestrians at varying scales, during occlusion and at unconstrained pose and orientation, the use of such techniques are not considered by the author to be robust enough to deal with real-world scenarios, and as such are not adopted in the proposed pedestrian detection technique.



(a) (b)

Figure 2.6: Hierarchical features; (a) 13 sub-windows [10]; (b) 6 sub-windows [11].

2.3.6 Multi-Cue Based Approaches

To increase reliability and reduce false-positives and false-negatives, some systems integrate multiple cues such as skin colour [121, 73, 49, 86, 117], face [73, 122, 120], or multiple body part detectors [120, 117]. For example, in [73] the authors initiate tracks on objects if a detected face within the object overlaps with skin regions. Other techniques, such as [120, 117] use several independent body part detectors, such as face, head, torso and hand detectors, whereby a human body is detected if multiple body part detections are made and their geometric relationships are consistent with a human body.

In general, however, the use of face and skin detection in pedestrian detection techniques reduces the applicability of the approaches to viewpoints where skin colour segmentation or facial detection may be performed [5]. For example, the approach in [122], which incorporates the face detection module of [119], has only been configured to detect frontal faces at distances ranging from 0.5 to 2.5 meters. In addition, weak classifications based on skin colour are also very sensitive to illumination changes [102]. Other techniques, such as [49], introduce further limitations as specific types of skin detection are employed, whereby a face is detected if skin and lip colour are detected within the same region.

2.3.7 Pedestrian Detection via Temporal Information

The pedestrian detection techniques presented so far in this chapter make their classification from single image frames. However, if accurate pedestrian tracking can be obtained, then this allows the use of temporal information, gained over a number of frames, to classify a foreground object as being a pedestrian. In addition, temporal information can also be employed to refine hypothesis about a pedestrian location in order to increase robustness and to decrease the computational cost

of the algorithm, e.g. if the algorithm runs at a relatively high frame rate, then a pedestrian detected in one frame is likely be situated in a similar position, pose and scale in a subsequent frame.

Many of these techniques apply rhythmic features, such as the periodicity of human walk, or motion patterns unique to human beings, such as learnt gait, to achieve their objectives. Some of these techniques simply classify a region as being pedestrian if it has recurrent motion [40]. Other techniques obtain the frequency [123, 124], or period [37], of this motion and ensure that it lies within the range of a walking pedestrian. Alternatively, techniques such as [125, 126, 24, 108, 98, 39, 25] use the characteristic criss-cross swing motion of the legs of a laterally walking pedestrian.

These types of techniques have an advantage over previous, single frame techniques, as they can use temporal data to validate their hypothesis over a number of frames. This can increase robustness to false-positives, however it can also be seen as a disadvantage as identification is delayed for several frames. For example, in some techniques the time-frame where the pedestrian is tracked before classification must be large enough to encompass the full temporal extent of movement. This makes techniques of this type inappropriate for some time critical applications like pedestrian collision detection from moving vehicles [33]. In addition, there exist other major drawbacks of approaches that use temporal information for pedestrian detection[33, 102, 39], in general;

- They cannot detect pedestrians standing still;
- They require the person to be walking at a roughly constant speed;
- They require the segmentation of the pedestrian in each frame to be highly accurate and un-occluded, depending on the approach this may also mean that pedestrians wearing skirts, pedestrians partially out of the field of view of the camera or crowds of pedestrians cannot be segmented correctly;
- They require the person to be within the scene for a number of frames, therefore pedestrians that are only in the scene for a few frames cannot be classified correctly;
- They require a relatively high frame rate;
- Many require the person to be walking roughly fronto-parallel relative to the camera, reducing the applicability of the approach;

- They cannot detect pedestrians performing unconstrained and complex movement such as wandering around, turning, jumping, etc;
- The techniques that classify pedestrians based on periodicity have difficulties with *non-stationary* periodicity, i.e. periodicity that changes with time, which may occur if a person is walking and then starts to run.

A final issue, and arguably the most important facing these techniques, is that there is generally an assumption made that an object can be robustly segmented and tracked from within the video sequence. In unconstrained conditions this is generally not the case. Due to these assumptions these techniques, although decreasing false-positives, generally add little insight as how to segment pedestrians in challenging scenarios as required in this work.

2.4 2D Pedestrian Tracking

As was the case with pedestrian detection, the robust tracking of pedestrians in unconstrained environments can be problematic (see section 1.3.2) and as such many algorithms make assumptions to simplify this complexity. These assumptions generally incorporate; how a pedestrian moves between frames; a pedestrian's temporal position; the appearance of a pedestrian in consecutive frames. Examples of assumptions made include; the frame rate is high enough so that a pedestrian's position does not change significantly between frames [87]; a pedestrian's appearance remains consistent between frames [20]; a pedestrian has constant velocity [127, 39, 20, 128, 45]; the speed of a pedestrian usually changes gradually when the pedestrian desires to stop or start walking [88]. Some of these assumptions can be true regardless of specific scenario parameters, which include camera position and pedestrian flow density. However, others can be invalidated in unconstrained environments. For example, the constant velocity assumption is generally untrue and a pedestrian may change direction abruptly and unpredictably [129].

There are two broad types of methodologies for tracking algorithms. The first is a *continuous detect-and-track* technique, whereby pedestrian detection techniques, such as those described in the previous section, are employed in each separate frame. Each of the detected pedestrians is then matched to the previous tracks via some similarity measure. The second technique can be thought of as *single detect-and-track* techniques, whereby once the detection is made, an independent tracking scheme is employed. For these techniques the number of pedestrians in the scene and

knowledge of their appearance is known *a priori* the current image frame. This information is then used to segment the pedestrians in the current frame. In these types of techniques, unexplained foreground regions are usually considered to belong to a single *new* pedestrian.

As with pedestrian detection, there is no proven technique for robust pedestrian tracking, but there has been progress on techniques to track an arbitrary number of pedestrians simultaneously [3, 44]. However, as stated in [130], as the number of pedestrians increases, system performance degrades due to limited processing power. This leads to an increase in time latency between processed frames making it increasingly difficult to keep an accurate track of every pedestrian in the scene.

2.4.1 Continuous Detect and Track

Tracking methodologies of this type generally apply pedestrian detection techniques independently to each individual frame. Each of the detected pedestrians is then matched to the previous tracks via some similarity measure. Techniques to match tracks to current tracks tend to be either *winner-takes-all* [73], or schemes to obtain the best global assignments, such as the Hungarian method [122, 98] or other bipartite graph matching schemes [88, 89, 44, 125].

The similarity measure used for matching is usually obtained via comparison of some features of the track to the corresponding features of pedestrians in the current frame. However, the choice of feature(s) is not always clear. In general, features can be divided into two distinct groups; *geometric* and *visual* features. Most approaches tend to use more than one feature during tracking to increase robustness to ambiguous matches.

2.4.1.1 Geometric features

These type of features are extracted from the positional or shape information of a detected pedestrian. Many of the geometric features proposed in the literature are obtained from a pedestrian's bounding box. These include comparisons based on area [73, 44, 69, 37, 123], width [131], height [96], perimeter [20, 44], distance between box centroids [132, 73, 37, 123], distance between median pixels [46], area of overlap [131, 132, 79], and density [44, 42], which is the percentage of pixels inside the bounding box. In general, independent of the choice of feature, user defined thresholds are applied to remove very unlikely matches.

These bounding box features can be inherently noisy due to errors in the motion segmentation

techniques that created them. This noise is taken into account by the feature, applied in [56, 68], which is based on geometric shape cues. The feature used is the *principal axis* of a person that can be thought of as a vertical line of symmetry through a human body. This axis is determined by minimising the median of squared perpendicular distances between the foreground pixels and a vertical axis [56]. The foreground pixels corresponding to the human body are symmetrically distributed to either side of the axis, therefore the errors of motion segmentation, which are assumed to take the form of white noise, are also distributed symmetrically and its effects are reduced.

Although the principal axis is more robust to noise, the feature is still subject to distortion due to changes in the shape of the silhouette of a person which is typical during walking, or due to occlusion. For example, when occlusion occurs the principal axis can be displaced from its correct position [56]. Some features are inherently less affected by these problems, such as the features of [133, 62, 87, 96] whereby the position of the object to be tracked is given by the head region and not the whole body. This head region may be determined by vertical projection peaks, template matching or by the median position of the top section of a pedestrian's bounding box.

2.4.1.2 Visual features

The use of features based on visual cues, rather then geometric, provides an alternative approach that has the advantage of being relatively unaffected by pedestrian position and pose. This, however, can also be viewed as a disadvantage as two objects at opposite ends of an image might have similar visual characteristics and so can have high correlation scores. Therefore in general, visual features are augmented with a geometric cue that provides positional information to the tracking algorithm. For example, [122] weights the use of visual features in proportion to the distance between possible matches for tracks. If the distance is great, then mainly positional information is used for matching as the closer the distance the higher the weighting of visual features becomes. The colour feature can therefore be used to disambiguate certain configurations of people when spatial tracking is not enough.

These visual features can be as simple as the average grey-scale value [20] or average colour value [69] of the region. These scalar values however, do not take into account the variability of colour values within a region. This is overcome in [62] where a Sum of Squared Differences (SSD) metric between colour intensities of corresponding pixels in regions is applied. A similar approach is applied in [35], where the average intensity of a number of feature points distributed regularly along the principal axis is used, and in [96] where the mean intensity of foreground

pixels for each row of the bounding box is obtained. However, the accuracy of these approaches are highly dependent on the alignment of the regions. A more robust approach is applied in [46] where colour matching is achieved using average intensities from the upper, lower and middle parts of the body.

Other techniques, such as [14, 70], employ the use of a colour histogram model for each person to differentiate between individual pedestrians. In these approaches, the histogram is simply used to depict the proportion of pixels, within the foreground region of a person, which display specific colour intensities. The histogram can then be used to estimate probability densities in colour space. However, a key parameter for colour models of this class is the level at which to quantise the colour values, i.e the number of bins, n, to be used in the histogram. If n is too high, many bins can be left empty and the estimated density can be noisy. If n is too low, then the structure of the densities in colour space is eroded. Therefore, histograms are effective only when n can be kept relatively low and where sufficient data is available to accurately describe the overall colour density [134]. Examples of these approaches are [14, 122] which quantises each channel of the RGB colour space into 4 bins, resulting in $4^3 = 64$ histogram bins, [70] which quantises each channel into 16 values resulting in 4096 histogram bins, [131] which quantises the R and G colour components into 4096 bins. However, a disadvantage to appearance models based on histograms is that they do not capture the spatial relationships of colour intensity information. Therefore, a person wearing a red shirt and black trousers may be interpreted in the same manner as a person wearing a black shirt and red trousers [135].

2.4.2 Single Detect and Track

Techniques applying this methodology tend to make the assumption that a pedestrian can be detected, using a previously described technique, for one or a number of frames when they first enter the scene. The detected person can then be used to guide the detection process in future frames. For example, when [43] detects an isolated pedestrian, it creates a template that is then used to detect the pedestrian within a small search area in the subsequent frame. Techniques using ASMs [53] can apply a similar technique, whereby an ASM in the new frame is initialised in the predicted position of the person via a Kalman filter. This ASM is subsequently fit to the image using edge features, which greatly reduces the computational requirements of the pedestrian detection technique for subsequent frames. However, most single detect-and-track techniques in pedestrian tracking algorithms tend to employ the use of *appearance based models*, which are statistical

models describing the appearance of the person. These models can be seen as more sophisticated models of visual features than those that were presented in section 2.4.1.

These appearance models are created and refined when a person is in isolation, which in general means that they form a coherent, un-occluded foreground blob that is easily extracted using background subtraction techniques. They are then used to segment, or refine segmentation, in future frames, even in the presence of occlusion. This is generally achieved via robust *per pixel* classification. In the literature a large variety of appearance models have been proposed [136, 134, 57, 87, 137, 68]. Some of these augment colour models with a *probability mask* component that incorporates the shape information of the person [46, 5, 138, 79, 58, 33]. However, it should be noted that for a number of these techniques, if two tracks come together the algorithm may fail to allocate the pixels to the correct model because of similarities in appearance, and thus tracking may be lost [79].

A major issue with all these techniques is that they are based on the assumption that either the appearance models are easily obtainable [55, 134, 57, 87, 68] or that they are known *a priori* [139]. For most techniques, an assumption is made that a new pedestrian can be easily segmented via background subtraction for enough frames for the model to stabilise and accurately describe the colour of pixels within the full area of pedestrian movement. In addition, it is assumed that occlusion of a new pedestrian will not occur until after these appearance models have been obtained, and as such can be used to reason about the pedestrian during occlusion. However, in unconstrained conditions, this may not the case and as such these techniques generally need human guidance in difficult scenarios to obtain the required appearance models. For example, in [68] a manual selection is required for the cases when a person does not constitute an isolated blob. If this guidance is not supplied then if two people do enter the scene at the same time then they will modelled and tracked as one. Whereupon, due to two individuals, consisting of various moving parts, being incorporated into a single visual model, the foreground extraction can be erratic [79].

Many of these techniques also have issues with regards to how their appearance models are updated. As with background models in unconstrained environments, compensations for gradual and sudden illumination changes should be incorporated to the appearance models. In addition, due to the non-rigidity of pedestrians, the models may also need to be updated to cater for changing pose, scale or orientation. However, large changes in pedestrian pose, or occluded regions can cause incorrect updates. In addition, some techniques, such as [46], cannot cope with changes in scale, so a person walking away or towards the camera results in incorrect template updates. It

is acknowledged however, that solutions for some of these issues have been proposed that lead to an increase in robustness. For example, [140] scales the portion of the image characterised by a person blob to a fixed resolution and then updates the probability distribution for each pixel in the model. In addition, [141] introduces the idea of maintaining multiple templates for each person, whereby if there is a significant change in their shape, a new template is created and assigned to the same person. However, regardless of these issues, the updating of colour model distributions is still problematic, and mirrors some of the issues for updating models in background subtraction techniques (see section 2.2.1). For tracking techniques of this class, the segmentation results depend directly on the stability of the underlying models [55]. If the models are updated with a few misclassified pixels, then they may increase the spatial covariance of underlying models [57], thus generating more mis-classifications in subsequent frames. This can cause the technique to lose the object it is tracking over time. To counter these problems, a weighting function is applied in [122] to give more relevance to pixels near the central point of the region and thus reduce the influence of background pixels that might be incorrectly included.

2.5 Summary

In this chapter a wide variety of techniques were introduced and discussed for pedestrian detection and tracking using traditional 2D based techniques. Many of these techniques produce good results when presented with constrained scenarios that allow assumptions to be made about the camera parameters, environmental conditions, pedestrian flow density and pedestrian appearance. Unfortunately, few of these, if any, produce reliable results for long periods of time in unconstrained environments [21]. Recently, 3D stereo information has been proposed as a technique to overcome some of the issues inherent in robustly detecting pedestrians. The use of stereo information carries with it some distinct advantages over conventional 2D techniques [21, 102]:

1. It is powerful cue for foreground-background segmentation [76];
2. It is not significantly affected by sudden illumination changes and shadows [142][3];
3. The real size of an object derived from the disparity map provides a more accurate classification metric than the image size of the object;

[3]However, in general this statement makes the assumption that the Photometric Compatibility Constraint – see section 4.2.1 – is upheld within a proposed disparity estimation technique, and so changes in illumination are consistent between all images obtained from the stereo camera rig.

4. Occlusions of people by each other or by background objects can be detected and handled more explicitly;

5. It permits new types of features for matching person descriptions in tracking;

6. It provides a third, disambiguating dimension for geometric features in tracking.

The idea of applying 3D range information was already presented in this chapter, in section 2.2.1, for use in a background range model. Some pedestrian detection algorithms described in this chapter, such as [5, 73, 74, 75], make use of such a background model, based on stereo vision disparity information. However, other techniques within this chapter also make use of stereo range data for a variety of other purposes. For example, it can reduce the computational cost associated with the template matching approaches of section 2.3.2. If the knowledge of how far an object, or blob, is away from the camera is available, then a pedestrian template can be scaled appropriately, reducing computational expense. In addition, a hypothesis generated from other techniques can be validated using depth information to filter out false detections, for example the size of a hypothesised person in Euclidean co-ordinates can be obtained and rejected if this size is too large or too small.

It should also be noted that important 3D information can also be obtained from a single camera using a 2D projective transformation between the image plane and a real-world groundplane – see section 4.3.2. This transformation can be seen as a mapping from the 2D image co-ordinates to 2D real-world groundplane co-ordinates. In addition, using a 3D rotation matrix and a translation vector, which can be obtained during calibration, a transformation between 2D image co-ordinates and 3D groundplane co-ordinates is also possible. This information can then be applied, as before, to scale models. Such a technique is employed in [58] to scale its 3D model. In their approach, the bottom of a foreground blob is assumed to be on the groundplane, therefore the 3D position of the bottom of the foreground blob can be estimated and hence the model can be scaled to the appropriate size. This technique, however, is subject to background segmentation errors. A similar approach is also applied in [143], which in conjunction with 3D stereo techniques, is used to refine the bottom of the bounding box to cover the entire shape of the pedestrian. Finally, [68] uses groundplane information between two cameras to refine the point of contact between the groundplane and the principal axis of a person. In each image the principal axis is obtained of each person, the principal axis from one image is warped to the second image plane using a 2D projective transformation and the point of intersection between the two axes is defined as the

groundplane point. Using this technique can be advantageous for tracking as a groundplane point can be obtained as a tracking feature, even when the point of contact is occluded from view.

Clearly, the use of 3D stereo to guide 2D pedestrian detection techniques carries some distinct advantages over conventional approaches. These approaches are examined in more detail in the next chapter, which also covers areas of projective and camera geometry, which are the basis for the stereo reconstruction of a 3D scene from two or more 2D images. In addition, the following chapter introduces various other pedestrian detection techniques that are based more firmly on the application of recovered 3D information, some of which are applied as the basic building blocks used by the author in the proposed pedestrian detection algorithm.

CHAPTER 3

Augmenting Pedestrian Detection and Tracking with 3D Information

3.1 Introduction

The potential advantages of augmenting 3D stereo information with traditional 2D image processing techniques are outlined in section 2.5. For pedestrian detection, these include robustness to illumination conditions, the ability to obtain the real Euclidean size and distance of an object, and advantages in the areas of occlusion reasoning. Pedestrian tracking can also take advantage of 3D information. Ambiguities between previous tracks and pedestrians detected in the current frame can be reduced using new 3D features, such as an object's Euclidean size or position. However, the application of 3D stereo information does not solve all problems posed by pedestrian detection and tracking techniques. For example; pedestrians may still appear in multiple poses, sizes and orientations; depending on the camera position and orientation, regions of a pedestrian's body may be at multiple depths and depth may not vary smoothly; and different objects, both foreground and background, can co-exist in very close proximity at similar depths making segmentation difficult. In addition to these problems, obtaining robust stereo information is non-trivial and can result in noisy or erroneous disparity maps, especially within areas of homogeneous colour.

Of course, depth information from a stereo camera rig is not the only way to obtain 3D information from a scene. For example, the use of laser-scanners, or laser radar (LADAR), has been employed to obtain 3D information for pedestrian detection in automated driving applications [144, 145], or to obtain 3D background models for pedestrian detection from mobile robots [146]. However, compared to active sensors of this nature, stereo vision provides a much higher

spatial resolution. Furthermore, being passive, there is little potential for interference with the environment [147].

In this chapter, an overview of 3D stereo vision pedestrian detection and tracking techniques is presented and subjectively evaluated in order to highlight respective strengths and weaknesses. Before this review section, a brief overview of single camera and multiple view geometry is presented as these geometries underpin stereo vision techniques and allow depth information to be inferred from two or more images.

3.2 Camera and Multiple View Geometry

In this section, a brief overview of *perspective projection*, or central projection, which describes the 2D image formulation by a pinhole camera, and *multiple view geometry*, which underlies the 3D reconstruction process from a stereo camera rig, are given. This review is not meant to be a thorough examination of the underlying geometries and the 3D stereo reconstruction process. Rather it is intended to provide the reader with enough insight into the relevant techniques as used in this work as well as the inherent associated difficulties. These include the disparity estimation techniques of chapter 4 and the 3D techniques used by the pedestrian detection algorithms in this chapter. For a more thorough discussion of projective and multiple view geometries, readers are referred to [148, 149, 150, 12, 151].

This overview is presented in three sub-sections; the first concentrates on projective geometry that describes the linear transformation from a 3D scene onto a 2D image plane; section 3.2.2 gives an overview of stereopsis, which attempts to reverse this transformation and infer the 3D structure and distance of a scene from two or more 2D images taken from different viewpoints; finally, section 3.2.3 examines the advantages in matching accuracy that can be exploited when using three or more cameras in a stereo rig.

3.2.1 Single Camera

A camera can be defined as a linear transformation between the real-world 3D Euclidean space to a 2D affine image space [148]. However, as both Euclidean and affine geometries are subsets of projective geometry [150], it can also be stated that a camera performs a linear transformation

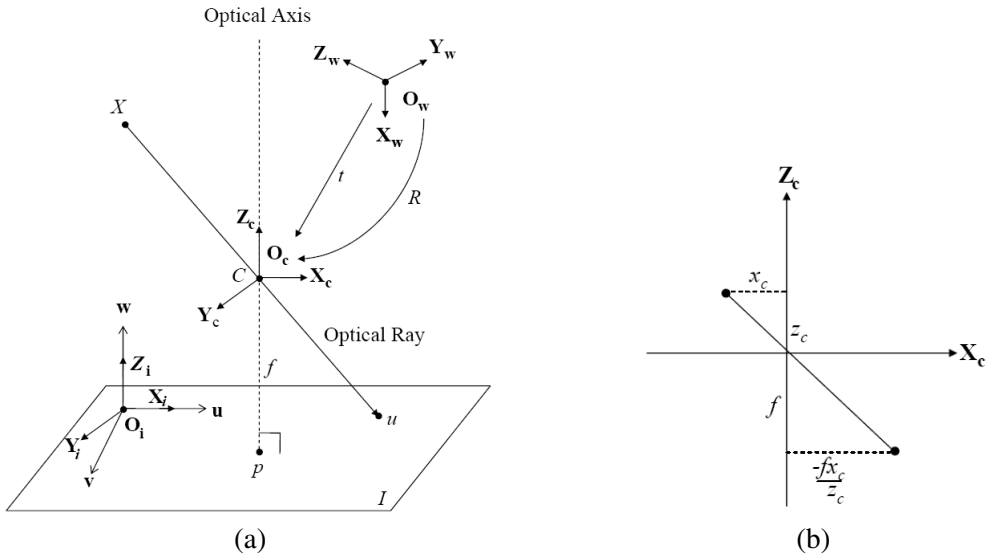


Figure 3.1: (a) The geometry of the perspective camera [12]; (b) Point in the image Euclidean co-ordinate system [12].

from the 3D projective space, $\mathbb{P}^3$, to the 2D projective space, $\mathbb{P}^2$, [12] according to

$$u = PX \tag{3.1}$$

where $u \in \mathbb{P}^2$, $X \in \mathbb{P}^3$, and P, called the *projective matrix* or *camera matrix*, is a 3×4 matrix that encapsulates both the intrinsic and extrinsic parameters for a given camera. An overview of this linear transformation, i.e. the perspective projection of a point in $\mathbb{P}^3$ onto an image plane in $\mathbb{P}^2$, is now given.

3.2.1.1 Basic Pinhole Camera

Throughout this section, an assumption is made that the camera model applied is that of a pinhole camera, which is the simplest camera model [148]. This pinhole camera model is used in most stereo vision based approaches, including the techniques proposed by the author. The basic pinhole camera can be defined as having an infinitely thin and small lens, i.e. the lens is a single point in 3D space, which allows exactly one light ray to pass through a single point on the image plane, the pinhole, and some scene point [149].

In a Euclidean framework, the basic pinhole camera maps 3D points X to 2D image points u using perspective (or central) projection, see figure 3.1(a), which can be fully described by X, the *camera centre*, C, and the *image plane*, I [152].

Using figure 3.1(a) as an illustrative example, the following definitions can be made:

Image (or Retinal) Plane The plane, I, onto which the image of the real-world is projected.

Focal Point The centre of the lens of the camera, C, or the position of the pinhole for the camera, with respect to the *camera's* co-ordinate system.

Camera (or Optical) Centre The centre of the lens of the camera, C, or the position of the pinhole for the camera, with respect to the *world's* co-ordinate system.

Optical (or Principal) Axis The line that runs through the focal point and is perpendicular to the image plane.

Principal Point The point, p, where the image plane and the optical axis meet.

Focal length The distance along the optical axis between the principal point and the focal point.

It can also be seen in figure 3.1(a) that in the image formation process, there are four distinct co-ordinate systems.

World Euclidean Has its origin at $\mathbf{O_w}$.

Camera Euclidean Has its origin at $\mathbf{O_c}$, $\mathbf{Z_c}$ is aligned with the optical axis by applying a rotation R and translation t from $\mathbf{O_w}$.

Image Euclidean Has its origin at $\mathbf{O_i}$, with $\mathbf{X_i}$ and $\mathbf{Y_i}$ aligned to the image plane, and $\mathbf{Z_i}$ parallel to the optical axis.

Image Affine Has its origin at $\mathbf{O_i}$, with 3 axes $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$. The reason this co-ordinate system is necessary is that pixels are not necessarily square, i.e. they may be sheared and their axes may not be scaled the same. If the pixels are *not* sheared then axes $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$ are aligned with $\mathbf{X_i}$, $\mathbf{Y_i}$ and $\mathbf{Z_i}$ respectively, if the pixels are sheared then axes $\mathbf{v}$ and $\mathbf{w}$ are usually aligned with $\mathbf{Y_i}$ and $\mathbf{Z_i}$ respectively [12].

3.2.1.2 Camera Extrinsic Parameters

A 3D point in the world Euclidean co-ordinate system, $X_w = (x_w, y_w, z_w)$, must undergo three transformations to be mapped from X_w to u. The first, defined by the camera extrinsic parameters,

transforms the point X_w into camera Euclidean co-ordinates, X_c, via [12]

$$X_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = R(X_w - t)$$

where R is a 3×3 rotation matrix and t is a 1×3 translation vector, which is the offset from $\mathbf{O_w}$ to $\mathbf{O_c}$. R and t are known as the 6 extrinsic parameters of a camera whereby 3 each are required for the rotation and translation.

3.2.1.3 Camera Intrinsic Parameters (Camera Calibration Matrix)

The second transformation projects X_c onto the image plane and obtains the resultant point in the 3D image Euclidean co-ordinate system, X_i. This transformation is obtained using basic trigonometry – see figure 3.1(b) – via [12]

$$X_i = \begin{bmatrix} \frac{-fx_c}{z_c} \\ \frac{-fy_c}{z_c} \\ -f \end{bmatrix}$$

The final transformation is to obtain U_i as a point, u, in the 2D image affine co-ordinate system. If the principal point in the image affine co-ordinate system is defined as $p = [\mathrm{u}_0, \mathrm{v}_0, 0]^T$, then the projected point can be defined in the image affine co-ordinate system in homogeneous co-ordinates as u, where

$$\begin{aligned} u = \begin{bmatrix} U \\ V \\ W \end{bmatrix} &= \begin{bmatrix} a & b & -\mathrm{u}_0 \\ 0 & c & -\mathrm{v}_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{-fx_c}{z_c} \\ \frac{-fy_c}{z_c} \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} -fa & -fb & -\mathrm{u}_0 \\ 0 & -fc & -\mathrm{v}_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{-x_c}{z_c} \\ \frac{-y_c}{z_c} \\ 1 \end{bmatrix} \end{aligned}$$

$$= K \begin{bmatrix} \frac{-x_c}{z_c} \\ \frac{-y_c}{z_c} \\ 1 \end{bmatrix}$$

where a is the scaling of the x axis, b is the shear, c is the scaling of the y axis and f is the focal length. In this equation, K is known as the *camera calibration matrix*. It has 5 degrees of freedom, which are the intrinsic parameters of the camera.

For most cameras the shear parameter b is zero. In realistic circumstances a non-zero skew might arise as the result of taking an image of an image, for example if a photograph is re-photographed [12]. Due to this, a true camera has only four internal camera parameters, since generally $b = 0$ [148]. If the assumption of square pixels are also made, i.e. the scaling of the x and y axes are the same, then there are only 3 degrees of freedom in K and it becomes

$$K = \begin{bmatrix} -f & 0 & -\mathrm{u}_0 \\ 0 & -f & -\mathrm{v}_0 \\ 0 & 0 & 1 \end{bmatrix}$$

It is assumed that this matrix K adequately describes the internal parameters of the cameras used in this work.

3.2.1.4 Projective Matrix

Returning to equation 3.1, if X_w is a point in the world Euclidean co-ordinate frame and u is a point in the affine image co-ordinate plane, and as $u = PX_w$, the equation can be rewritten as

$$\begin{aligned} u &= KR(X_w - t) \begin{bmatrix} X_w \\ 1 \end{bmatrix} \\ &= [KR| - KRt] \begin{bmatrix} X_w \\ 1 \end{bmatrix} \end{aligned}$$

Therefore

$$P = K[R| - Rt] \tag{3.2}$$

where P, called the *projective matrix* or the *camera matrix*, is a 3×4 matrix that encapsulates both the intrinsic and extrinsic parameters for a given camera. If the projective matrix of a given camera is known then that camera can be defined as *fully calibrated*.

3.2.2 Two Cameras - Stereopsis

As seen, a camera performs a linear transformation from $\mathbb{P}^3$ to $\mathbb{P}^2$. 3D stereo vision, however, is interested in reversing this process and obtaining the 3D structure of a scene from two or more 2D input images. In this section, an overview of this process is presented using a stereo rig with 2 fully calibrated cameras, see figure 3.2(a). In this figure, the two camera centres, image planes and focal lengths are labelled as C_1, C_2, I_1, I_2, f_1, and f_2 respectively. Notice in these figures the image plane is *in front* of the camera centre, as opposed to the previous camera diagrams in figures 3.1(a) and (b). This difference is equivalent to having the image plane behind the camera centre and reflecting the image x and y axes [152]. In these figures the switch has been made solely for convenience of illustration.

The basic premise behind stereo reconstruction is illustrated in figure 3.2. For a single point u_1, on I_1, a unique 3D ray, L_1, can be determined that passes through u_1 and the camera centre C_1, see figure 3.2(b). Similarly, given an image point u_2 on I_2 a second 3D ray, L_2, can be defined, see figure 3.2(c). Using these two rays, if u_1 and u_2 are projections of the *same* 3D point X, then L_1 and L_2 should intersect in 3D space at this single point, X. The 3D co-ordinates of X can then be calculated using trigonometry in a process is called *triangulation*, see figure 3.2(d). It is important to note that unless there is knowledge of the Euclidean distance between the two camera centres C and C_2, known as the *baseline* B, see figure 3.2(d), then the 3D reconstruction can only by up to a scale factor, i.e. there exists an unknown scaling transformation between the 3D reconstruction and the real-world 3D scene.

3.2.2.1 Epipolar Geometry

The difficulty with this approach is that, in general, given u_1 the corresponding point u_2 is *not* known *a priori*. Therefore, before X can be obtained as a point in 3D space, a search must be undertaken to find the match, or corresponding point, to u_1. This is the main challenge associated with stereo vision approaches, as finding a match can be non-trivial. A review of techniques used to obtain these matches is presented in section 4.2.

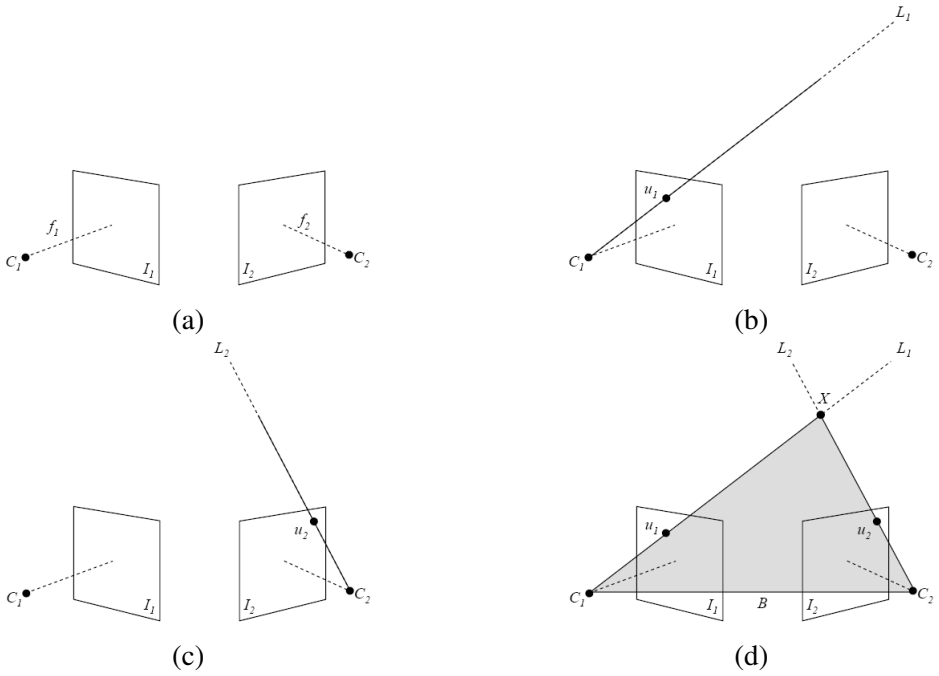


Figure 3.2: Stereo reconstruction; (a) Stereo rig; (b) Ray L_1; (c) Ray L_2; (d) Triangulation.

This search can, however, be narrowed from a 2D image search to a 1D line search using geometric constraints. In figure 3.3(a) it can be seen that although the exact point where X exists in 3D space is unknown, it is known that X *must* lie somewhere on the ray L_1. This is illustrated in figure 3.3(a), where three possible matching locations of X on L_1 are labelled as X^1, X^2 and X^3. It can also be seen from this illustration that each different position results in a different corresponding point in I_2, namely u_2^1, u_2^2 and u_2^3 for the 3D points X^1, X^2 and X^3 respectively. An important constraint on these points is that as X^1, X^2 and X^3 are collinear in $\mathbb{P}^3$ then the points u_2^1, u_2^2 and u_2^3 *must* be collinear in $\mathbb{P}^3$ as a projective transformation, by definition, maps lines onto lines (or points) [151]. The line that passes through u_2^1, u_2^2 and u_2^3 is known as an *epipolar line*, l_2^1, see figure 3.3(b), and is the projection of L_1 onto the image plane I_2. In addition, the epipolar line also passes through the image of C_1, known as the *epipole*, e, of C_1, and the image of the required 3D point X, see figure 3.3(c). Figure 3.3(d) shows the epipolar lines for two distinct points on I_1, notice how each epipolar line intersects the baseline at the epipole, e, of C_1.

Fundamental Matrix The epipolar constraint can be fully described by a single matrix, known as the *fundamental matrix*, F, which encodes the epipolar geometry between a pair of cameras [12]. Assuming that C_1 is aligned with the world origin, $F = (K_1^{-1})^T.t \times R^{-1}K_2^{-1}$ [12], where K_1 and K_2 are the camera calibration matrices of the two cameras, R is the 3×3 rotation matrix between the two cameras local co-ordinate systems, t the translation between C_1 and C_2, and .

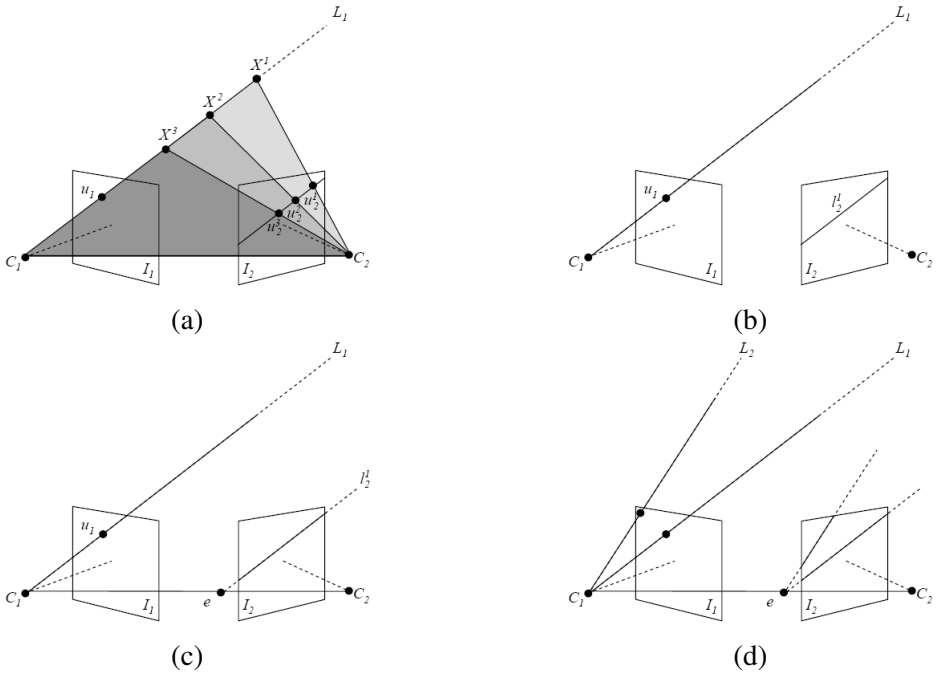


Figure 3.3: Epipolar geometry; (a) Epipolar constraint; (b) One epipolar line; (c) Two epipolar lines; (d) Epipole.

and $\times$ are the dot and cross product respectively. Using F, the relationship between u_1 and l_2^1 is given by

$$l_2^1 = Fu_1 \tag{3.3}$$

In addition, F has the property that the transpose of F, namely F^T, describes the epipolar constraint in the other direction, so that given u_2 the corresponding epipolar line in I_1 can be obtained via;

$$l_1^2 = F^T u_2 \tag{3.4}$$

3.2.2.2 Image Rectification

A special stereo camera rig setup exists, called a *canonical configuration*, which simplifies the overall process for 3D reconstruction. In this setup, the baseline is aligned to the horizontal (or vertical) co-ordinate axis, the optical axes of the cameras are parallel, the epipoles move to infinity, and the epipolar lines in the image planes become parallel [12], see figure 3.4(b). There are three main advantages to having a canonical camera setup; firstly the search for matching points is simplified by the simple epipolar structure; secondly a neighbourhood correlation-based match-point

search can succeed, because local neighbourhoods around matching pixels will appear similar and hence will have high correlation [153]; and finally the canonical setup reduces the complexity of the 3D reconstruction process.

Unfortunately, most camera setups are not canonical, see figure 3.4(a) where θ_1 and θ_2 are not at 90 degrees to the baseline. However, a processing step called *rectification* can be employed to make a more general stereo rig setup canonical. A popular rectification process, called *planar rectification*, can be implemented by projecting the original images onto a new common image plane that is parallel to the baseline [149].

The idea behind planar rectification is to define two new projective matrices, $\tilde{P_1}$ and $\tilde{P_1}$, obtained by rotating the old projective matrices, P_1 and P_1, around their optical centres, C_1 and C_2 respectively, until the image planes becomes coplanar, thereby moving the epipoles to infinity and forcing the epipolar lines to become parallel. In addition to this rotation, the internal parameters of K_1 and K_2, such as the focal lengths, can be altered slightly in each matrix so that they are made equal, e.g $f_1 = f_2$. This can be implemented so that the epipolar lines become parallel *and* collinear, such as in figure 3.4(b), or to remove distortions such as shear.

In effect, the process of image rectification re-samples the original images so that they appear to come from two cameras with projective matrices $\tilde{P_1}$ and $\tilde{P_2}$. This re-sampling is achieved using a pair of 2D projective transformations, or homographies, $\tilde{H_1}$ and $\tilde{H_2}$, whereby each homography encodes the required rotation and translation of the image plane. The rectified images can be thought of as acquired by a new stereo rig, obtained by rotating the original cameras [154].

The advantages of this rectification approach are that it is mathematically simple, fast and, due to the projective transformation, it preserves image features such as straight lines [155]. Using this approach only two 3×3 transformations need to be stored, and only six multiplications, six additions and two divisions are needed per rectified pixel [12]. However, planar rectification techniques, such as [154], are not general approaches and a large forward component in the camera movement it may produce badly warped images [155]. In addition, many approaches do not guarantee an optimal solution whereby the amount of distortion in the re-sampled images is not guaranteed to be minimal. For many stereo vision pedestrian detection algorithms an assumption is made that the input images are rectified and distortions, such as radial distortion, are removed. In the proposed system of this thesis, the input images are either assumed to be pre-rectified or that the two homographies, $\tilde{H_1}$ and $\tilde{H_2}$, can be obtained using the technique outlined in [154].

(a) (b)

Figure 3.4: Rectification; (a) Unrectified images; (b) Rectified images.

3.2.2.3 Triangulation

The reconstruction of a 3D point from 2 point matches, u_1 and u_2, is simplified from a canonical stereo rig. Let $u_1 = (x_1, y_1)$ and $u_2 = (x_2, y_2)$ be two corresponding points in their respective 2D image affine co-ordinate systems, and let the principle points, p_1 and p_2, be the respective origins of the two co-ordinate systems. In addition, let C_1 be the world Euclidean co-ordinate system origin, $\mathbf{O_w}$; C_2 be offset from C_1 by some translation t; f be the common focal length after rectification; and B be the Euclidean length of the baseline between the camera centres, see figure 3.5(a). The position of the point $X_w = (x_w, y_w, z_w)$ in 3D world Euclidean co-ordinates can then be determined via *triangulation*.

From figures 3.5(a) and (b) it can be seen that the triangles (C_1, X_w, C_2) and (u_1, X_w, u_2) are proportional to each other, which leads to the equation

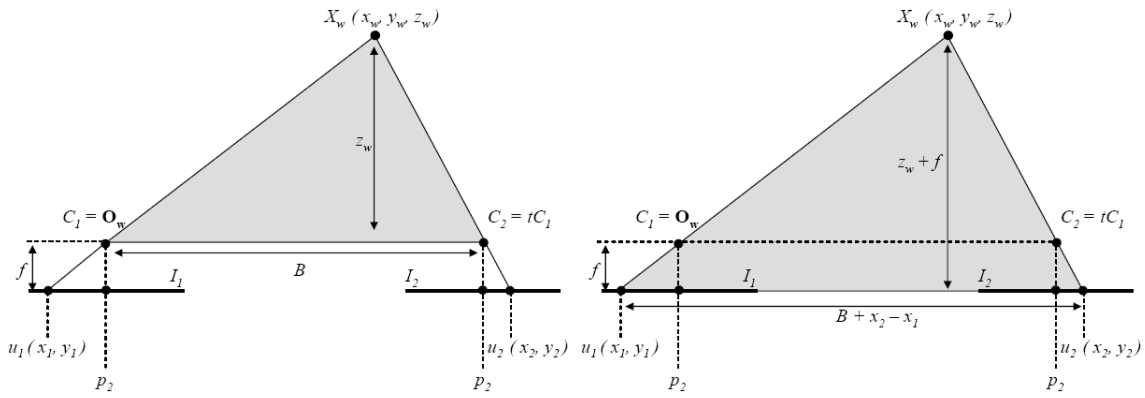
$$\frac{B}{z_w} = \frac{B + x_2 - x_1}{z_w + f}$$

It should be noted that f, x_1 and x_2 are defined in pixels, whereas B and z_w are defined by Euclidean distance. This mixture of measurement basis is acceptable as the extension to the lengths of the triangle (C_1, X_w, C_2) is via pixels in *both* axis, therefore the extension to the dimensions of the triangle (u_1, X_w, u_2) is kept within proportion. In addition, it should be noted that although f is an absolute value of pixels, x_1 and x_2 are the pixel *offsets* from their respective principle points, therefore the distance between u_1 and u_2 is $B + x_2 - x_1$ and not $B + x_2 + x_1$.

Solving for z_w the equation for the depth of X_w is obtained as

$$z_w = \frac{Bf}{x_2 - x_1} \tag{3.5}$$

The value of $x_2 - x_1$, which is the horizontal difference from where the point u_1 occurs in the left image and where the corresponding point u_2 occurs in the right image is known as the *disparity*. From equation 3.5 it can be seen that the value of the disparity is inversely proportional



(a) (b)

Figure 3.5: Triangulation; (a) Triangle (C_1, X_w, C_2); (b) Triangle (u_1, X_w, u_2).

to depth of X_w.

Finally, the values of x_w and y_w are obtained via

$$x_w = \frac{x_1 z_w}{f} \tag{3.6}$$

and

$$y_w = \frac{y_1 z_w}{f} \tag{3.7}$$

3.2.2.4 Occlusion

When obtaining the corresponding point to u_1 in I_2 it is important to note that although l_2^1 passes through the position where the corresponding point to u_1 *should* appear in I_2, the image of the point may not be within the bounds of the image plane or it may be *occluded*. Occlusion occurs when the projection of a 3D point X can be seen in one image, but the projection of the same 3D point is somehow obstructed in another image. This makes it impossible to reconstruct the 3D position of X from the two input images. Occlusion can occur for two reasons; *group occlusion* and *self occlusion* – see section 1.3.1.

In group occlusion, the point on an object X, which is projected onto u_1, is occluded from view in I_2 by a second object Y, see figure 3.6(a). This type of problem can cause *partial*, or in the case of this specific example, *full* occlusion of the object X from view in I_2 by the second object Y, see figure 3.6(b). Self occlusion is caused when specific points on an object, X, are occluded by *other* points on the same object, see figures 3.7(a) and (b) where X is a spherical

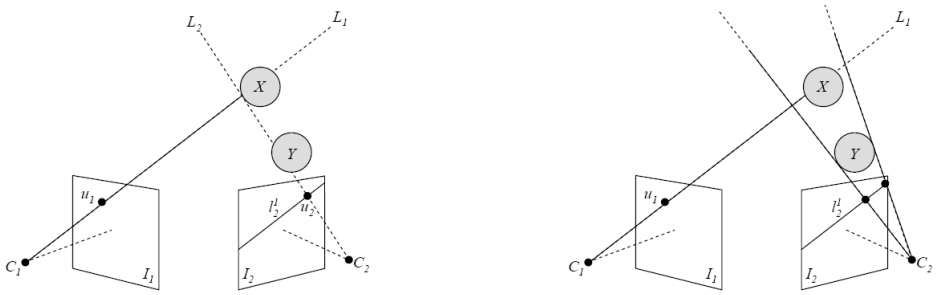


Figure 3.6: Group occlusion; (a) The point on object X, projected onto u_1, is occluded from view in I_2 by object Y; (b) The whole of the object X is occluded from view in I_2 by object Y.

object and the dark region on the sphere in each figure depicts the points on X that are visible from C_1 and C_2 respectively. Due to the offsets of the two cameras only a proportion of the pixels projected onto I_1 from X can be seen in I_2 and vice versa, see figure 3.7(c) where the dark region on the sphere depicts the area of the sphere that can be viewed from both C_1 and C_2. This means that *only* the points on X that are within this common region can be accurately matched between I_1 and I_2, leaving all other points on X either undefined or incorrectly matched.

Self occlusion is less of an issue (and generally ignored) for stereo camera rigs with smaller baselines, however as the baseline is increased the common viewing area on X decreases making stereo correspondence matching more problematic. A similar effect can be seen by figure 3.7(d) where, although the baseline stays the same, the object coming closer to the stereo camera rig causes the common viewing area on X to significantly decrease.

3.2.3 Three or More Cameras

If a third camera is added to the stereo camera rig then it becomes possible to reduce some of the ambiguities in stereo correspondence matching techniques. One such area from which ambiguities can be reduced is that of occlusion as the third camera can provide new data from a different, possibly complementary, position to the other cameras. In addition, a new geometric constraint can be employed to reduce correspondence matching ambiguities called the *trinocular constraint* [12]. It has been shown that applying this constraint provides a significant advantage in matching accuracy over binocular stereo at the local matching stage [156], although it is slightly more computationally expensive.

The addition of more cameras does not introduce any further geometric constraints, as there is no relation involving four or more cameras that cannot be factorised into relations of fewer

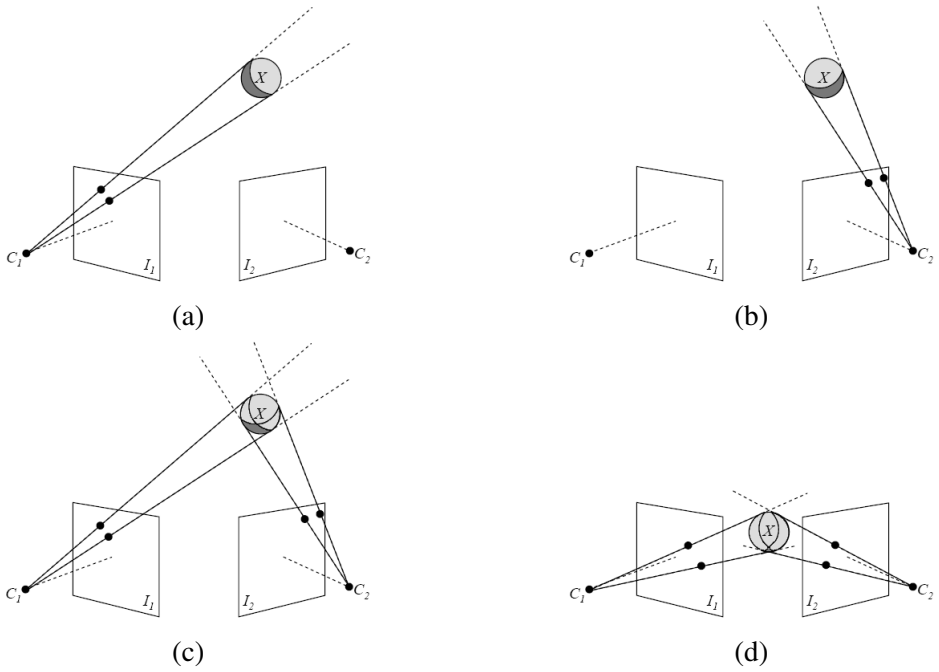


Figure 3.7: Self occlusion; (a) Object points visible from C_1 (in dark grey); (b) Object points visible from C_2; (c) Object points visible from C_1 and C_2; (d) Closer object has less visible common points.

cameras [157]. However, using more cameras has advantages in other areas, such as occlusion. For techniques using N cameras it is advantageous to use a wide-baseline system as if object X is occluded in I_1, then the greater the value of N the more likely it becomes that the same object will *not* be occluded in one of the other images. This is illustrated in figures 3.8(a)-(d) where the projection of X is occluded by Y, or vice versa, in I_1, I_2 and I_3, but not in I_4. In addition to this advantage, in general the wider the baseline, the more accurately 3D reconstruction can be achieved as it is less affected by calibration errors.

However, such rigs have a number of disadvantages, including an increased susceptibility to self-occlusion problems. For example, in figure 3.8 there is little overlap in common viewing areas between any two cameras and for some camera pairs, such as I_1 and I_3, there are none at all. For this reason, these approaches generally do not use feature similarities between pixels to obtain the 3D position of points, instead *shape-from-silhouette* based approaches are normally employed which obtain the 3D shape and position of *regions* and not individual points. An overview of this technique can be seen in figure 3.9. In these approaches, the silhouettes of foreground regions are obtained in each of the images using background subtraction techniques – see figure 3.8. These silhouettes are then back-projected from each of the camera centres and logically ANDed together, see figures 3.9(a)-(d) where it can be seen how the shape-from-silhouette is refined with

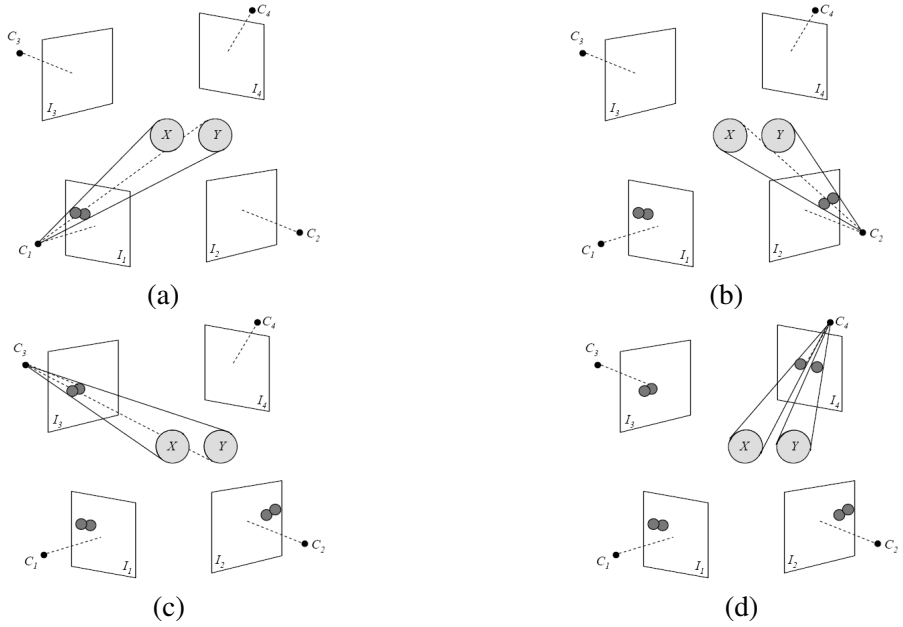


Figure 3.8: Projections of objects X and Y onto; (a) I_1; (b) I_2; (c) I_3; (d) I_4.

each additional camera added. However, it should be noted that if there are not enough cameras in the setup then the reconstructed 3D object shape may be perturbed – see figure 3.9(d) – or multiple objects may become merged as in figures 3.9(a) and (b).

Pedestrian detection techniques that employ these wide-baseline techniques tend to need a high concentration of cameras to monitor a relatively small spatial region that are normally positioned in an approximate circle around the region of interest. For example, [158] uses 5 cameras to monitor a $2 \times 2 \times 2$ metre area, [159] uses 5 wide angle lens cameras to cover a 5×4 metre room and [137] uses 16 cameras to cover a 3.5×3.5 metre area, 8 of which are at a lower height level and the other 8 cameras are located directly on top of them at a higher level. Techniques that require this high concentration of cameras are expensive to install, calibrate, maintain and are generally not scalable to larger surveillance areas. Therefore, these approaches are not considered viable for scalable pedestrian detection and do not consider them further in the review of 3D pedestrian detection and tracking techniques presented in this chapter.

3.3 Post-processing Stereo Data for Pedestrian Detection

Techniques using short-baseline stereo camera rigs tend to set and align the *camera* Euclidean co-ordinate system of a particular camera, say C_1, to that of the *world* Euclidean co-ordinate system, meaning that $\mathbf{O_w} = C_1$, $\mathbf{X_w} = \mathbf{X_{c1}}$, $\mathbf{Y_w} = \mathbf{Y_{c1}}$ and $\mathbf{Z_w} = \mathbf{Z_{c1}}$. Correlation based approaches, see section 4.2, are then employed to obtain the disparity of pixels for pairs of input images.

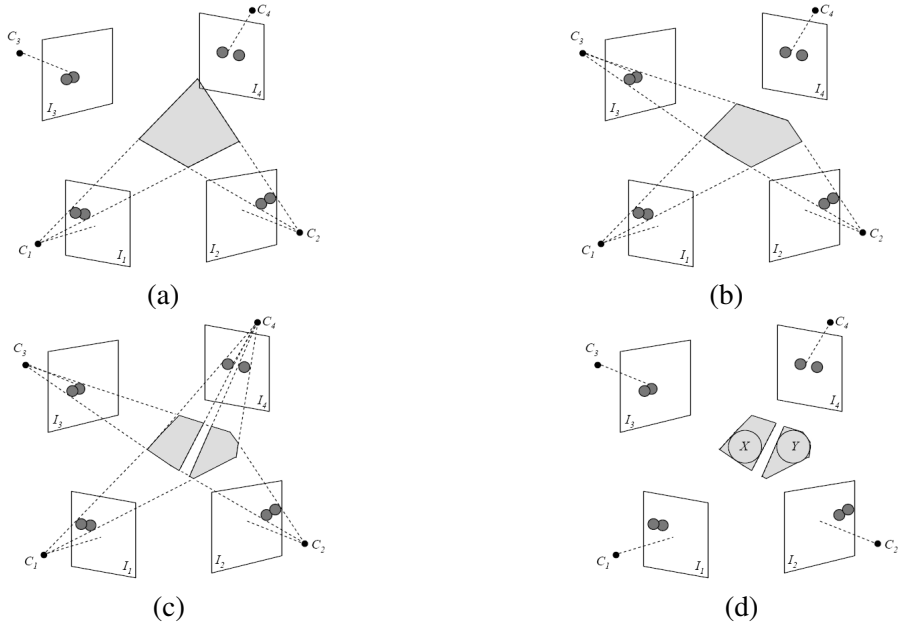


Figure 3.9: Shape-from-silhouette; (a) Back-projection from C_1 and C_2; (b) Back-projection from C_1, C_2 and C_3; (c) Final back-projected objects; (d) Final and original objects overlaid.

Pedestrian detection techniques that use stereo information tend to then use both the visible spectrum input image, I_1, and either the disparity information of I_1 or the 3D points obtained from these disparity values. For techniques that use disparity information a post-processing step is generally employed that quantises the disparities of each pixel in I_1 into monochrome colour values which are then set in a *disparity image* that has the same dimensions as that of I_1. For example, the depth within the synthetic scene of figure 3.10(a) ranges between 0 and 23.48 meters. Figure 3.10(b) shows the disparity image of this scene, where the brighter the pixel colour, the greater the disparity and the closer the point is to the camera.

The stereo information used by pedestrian detection techniques can be post-processed using a number of techniques designed to either; (a) reduce computational expense; (b) remove background regions; or, (c) cluster the foreground into likely pedestrian candidates. This post-processing can involve; removing all points that are outside a 3D volume of interest [143, 3]; using a background disparity model to obtain foreground depth regions [5, 73, 74, 75]; using connected components to create regions of slowly varying disparity [73]; or segmenting the disparity image into a number of *depth layers* [26, 74, 75].

Some of these techniques can be prone to difficulties under certain scenarios; background disparity models cannot correctly segment objects that are at the same depth as background regions, as was discussed in section 2.5; obtaining depth layers can segment a pedestrian into two

(a) (b)

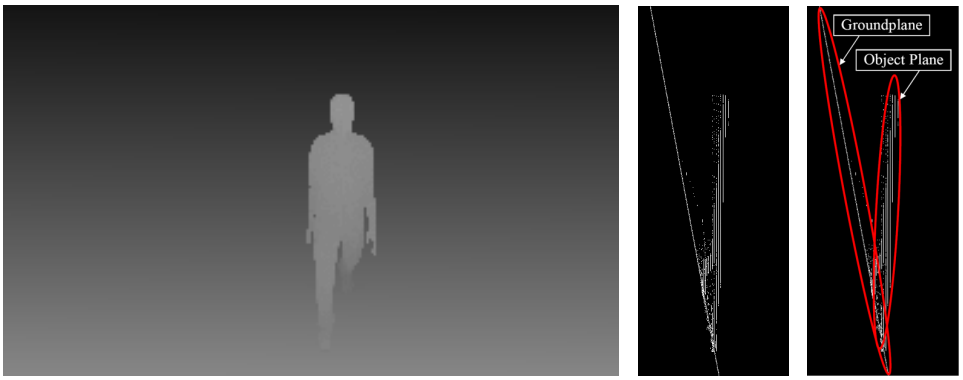
Figure 3.10: Synthetic test data; (a) Visible spectrum image; (b) Disparity image.

(a) (b) (c)

Figure 3.11: Disparity layers; (a) Layer 1; (b) Layer 2; (c) Layer 3.

regions as a person's body can be at multiple depths from the camera, see figures 3.11(b) and (c); and connected component analysis can cluster two or more pedestrians at the same depth into the same region if they are connected by smooth disparity values, such as the ground in figure 3.11(b). As such, these techniques are not adopted by the pedestrian detection approach proposed in this thesis.

A post-processing approach, similar to defining a 3D volume of interest (VOI), is proposed in [160] and used by a number of pedestrian detection approaches, such as [161, 1, 162, 163]. The technique, however, does not require a user defined VOI, instead it removes all points that are positioned on or below an automatically detected groundplane. The technique is based on construction of a *v-disparity* image, where v is the ordinate of a pixel in the (u, v) image co-ordinate system [160]. A v-disparity image is of size $n \times Y$, where Y is the height of the original disparity image D_1, and n is the maximum disparity within D_1. A pixel (n, y) in the v-disparity image represents the number of pixels in the row y of D_1 that has a disparity value of n. This number is then encoded into the v-disparity image as a grey-level value, see figure 3.12(b) which shows the v-disparity image for a disparity map containing a flat groundplane and a single person. In the v-disparity image a flat groundplane is mapped to a single bright line segment, see figure 3.12(c), which can be extracted using a Hough transform [160]. The detected groundplane can be differentiated from all other objects in the image using the knowledge that all non-groundplane objects that are present in the scene are mapped to *vertical* bright segments [1] in the v-disparity, whereas the groundplane line exhibits a slope within a certain predefined range, see figures 3.12(c),



(a) (b) (c)

Figure 3.12: V-Disparity; (a) Single person and groundplane disparity map; (b) V-Disparity of (a); (c) Object planes of v-disparity of (a).

(e) and (g). Once the profile of the road has been extracted, the disparity values on the road surface are known for each value of y, therefore if the disparity of a pixel is greater than the disparity of the road plane for a given y, then it belongs to a possible foreground object, otherwise the point is removed as background.

This post-processing technique has the advantage of being able to detect and remove the groundplane, even within highly cluttered scenes. However, the process of segmenting the groundplane is more complex if the ground is not exactly planar, in which case a piecewise linear curve must be extracted using the Hough transform [160], or when accurate disparity is difficult to obtain, for example if the groundplane is homogeneous in colour. In addition, the technique is highly dependent on the X-axis of both the stereo rig and the groundplane being parallel (i.e. the cameras in the stereo rig are rectified and at the same height above the groundplane). If the camera rig does not have this setup significant noise is introduced to the v-disparity image. In the proposed pedestrian detection approach in this thesis, a VOI that incorporates groundplane information – similar to that applied in [11] – is defined via calibration, see section 5.2.1. However, in future work it is envisioned to incorporate a technique, similar to the v-disparity approach, for the automatic detection and calibration of the required groundplane information. However, the technique should be extended so that the camera setup does not have to strictly adhere to the requirements of a parallel stereo rig and the groundplane X-axis.

3.4 Pedestrian Detection using Stereo Information

In the literature, post-processed stereo regions (obtained, for example, using background subtraction and connected component analysis) have been used as as the basis of pedestrian detection classifiers. For example, in [13] it is assumed that a pedestrian is segmented into a single foreground connected component region, which is obtained using stereo and colour information. A region containing a possible pedestrian is then skeletonised and converted into a graph representation, see figures 3.13(a) and (b), which is subsequently classified using a Support Vector Machine (SVM).

Other techniques assume that a pedestrian constitutes one or more foreground regions. The fragmentation of pedestrians into multiple foreground regions can occur for a number of reasons. These include issues with background disparity models if sparse disparity estimation techniques are employed or a lack of texture information is present within areas of the input image, causing erroneous disparity information to be estimated. In addition, issues with traditional 2D colour intensity background models may result in the over-segmentation of a foreground pedestrian due to a lack of intensity variation between foreground and background objects.

For techniques that assume a pedestrian may be initially over-segmented into a number of foreground regions, or blobs, the general approach taken is to cluster a number of the initial regions together to form an individual person. For example, in [76] foreground blobs are grouped if they have similar disparity values and the resultant grouped region does not exceed the size range of a normal person. A similar approach is adopted in [74], however it first filters the foreground blobs, discarding any that do not meet predefined criteria. They observe that a human body has distinctive 3D curvatures similar to a sphere (head), plane (body), and cylinder (legs). This information is employed to reject any foreground connected component region, from a particular depth layer, which has a non-convex 3D structure. This is achieved by fitting a parabolic curve across the disparity values in each row of the foreground region and the fitting curve was then checked to be either convex or concave with respect to the camera rig. All concave regions are then discarded.

An alternative approach for clustering foreground regions is taken in [14]. People-shaped regions are created by searching through the space of possible groupings between foreground blobs to create a minimum spanning tree, i.e. a graph that spans all the blobs with the minimum number of edges, see figure 3.13(c). In this approach, all edges between blobs that are less than a certain length threshold are considered as viable groupings. In figure 3.13(c) the match on the

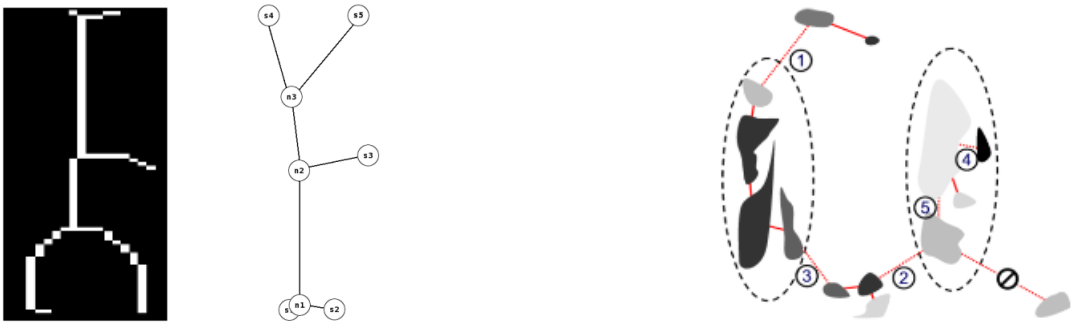
right hand side marked with a $\oslash$ is over this length threshold and so not considered a viable match. To obtain this graph the n longest links are chosen – see figure 3.13(c) where $n = 5$ – and used to generate 2^n hypothesised blob clusters. Each potential cluster is evaluated by obtaining the major and minor axis lengths from an ellipse surrounding the points for each blob cluster. If the axis lengths are too small then they are removed, otherwise they are compared with the *ideal* ellipse axis sizes of people. Of the 2^n hypothesised clusters, the one with the best match to these ideal values is chosen as the correct cluster – see figure 3.13(c).

Not all 3D-based approaches employ background subtraction techniques from disparity models. For example, the two camera wide-baseline system of [164] firstly uses a background intensity model to obtain foreground regions in each input image. Disparity estimation and triangulation is then undertaken *only* on these foreground points. The technique then clusters these foreground object 3D point clouds into regions using *mean shift clustering*. The approach presents good results for pedestrians at large distances, and therefore at low-resolution, from the cameras, however the technique is affected by shadows and background subtraction issues. In addition, their approach only obtains a 3D reconstruction up to an unknown scale factor and therefore the window size of their clustering kernel must be hand tuned so that over-/under-segmentation of pedestrians does not occur.

Other techniques, such as [11], do not use any type of background model. Instead the approach in [11] clusters all 3D points that lie within a 3D volume of interest (VOI), removing all 3D points that are close to the groundplane or above 2 metres in height. The technique then employs a *subtractive clustering* based approach to obtain the centre of 3D objects within the VOI, whereby each 3D point within the VOI is considered as a potential object centre. Using this technique a 3D spatial distribution, d_i, is calculated for each 3D point, p_i. The value of d_i represents the weighted number of 3D points that are contained within a pre-defined neighbourhood $1 \times 1.5 \times 1$ metre area to p_i, where points closer to p_i have a higher weight according to a Gaussian function. Candidate centres surrounded by a large number of points within the defined neighbourhood exhibit a high value of d_i. The point exhibiting the maximum density is selected as a cluster centre if it is above a pre-defined threshold. If a viable object centre is found then a subtraction process is implemented, whereby the density corresponding to the cluster centre becomes zero and all other 3D points in the neighbourhood of $1.5 \times 2.25 \times 1.5$ metres to the centre are decreased by an amount that is a function of the distance to the centre. This process is then re-iterated, selecting the new highest value d_i, until there are no more significant densities, or the ratio of the last obtained object centre

density to the new density is greater than a threshold. However, this technique is highly susceptible to the detection false-positives if background objects are present within the bounds of the VOI. If these background objects are static then they may always be detected as false-positives. To overcome this, [11] removes false-positives using a final classification on the objects from an SVM. However, problems with this technique occur when objects get too close to each other and can be well represented by a single Gaussian distribution, therefore the two may be subtracted at the same time. In addition, pedestrians that are highly occluded tend to have a lower density, due to a lower number of points being visible, so that if the occlusion is great then the value of d_i can drop below the required threshold needed for correct detection. In addition, for similar reasons, pedestrians at a large distance from the camera may have a density less than the required threshold, due to their smaller image size, and thus, a lower number of 3D points. Due to the clustering of 3D points, the two previous techniques of [164] and [11] do share some similarities with the pedestrian detection technique proposed in this thesis. However, compared to the proposed approach, [164] is prone to problems arising from background modeling in unconstrained conditions. While [11] can suffer from issues caused by the setting of the pre-defined threshold for point density.

However, it should be noted that full disparity estimation and 3D reconstruction does not have to be implemented to exploit information from a stereo camera rig. The technique proposed in [63] does not use this information, nor does it make use of colour models or shape cues of individual people. Instead, the authors use background colour intensity models and a *groundplane homography* constraint, which can be viewed as a 2D projective transformation that maps points on the groundplane in one image, to points on the groundplane in a second image – see section 4.3.2. The homography constraint implies that only pixels corresponding to the ground plane locations of foreground objects, such as peoples' feet, consistently warp to foreground regions in every view in the camera stereo rig. The premise of the technique is then to obtain foreground likelihood maps from each view, and warp them onto a reference view and fuse the information from the multiple cameras. The pixels pertaining to the feet of the people can then be segmented out using a user defined threshold and final regions can be created using connected component analysis. However, this technique has a number of disadvantages; if the gait of a person is such that the distance between the two feet is at a maximum, then two regions, one for each foot can be obtained, resulting in a false-positive detection; the homography can map non-groundplane regions onto other non-groundplane regions, meaning that a number of wide-baseline cameras at various angles to the scene need to be used to obtain robustness to false-positives; and the technique is



(a) (b) (c)

Figure 3.13: Stereo regions; (a) Skeletonised region [13]; (b) Region graph [13]; (c) Blob groupings [14].

susceptible to detecting false-positives due to shadows. This final issue can be seen in the fact that the approach has a number of similarities to the technique employed in [165], which is used to detect and remove shadows and low-lying regions on the groundplane.

3.4.1 Plan-view statistics

As was the case with 2D based pedestrian detection approaches – see section 2.3.1 – some 3D techniques adopt an overhead viewpoint for the camera stereo rig in order to minimise the occlusion between people that can occur with more oblique camera angles. In addition, as the spatial dimensions of most people fall within a limited range most of the time [15], the pedestrian model can be simplified allowing more generalised pedestrian models to be used. In [166] a stereo camera is mounted above a door, pointing downwards towards the groundplane with a view to counting shoppers as they enter or exit a retail environment. In the approach, background subtraction is not implemented, instead a 3D volume of interest (VOI) is defined that limits further processing to the head and upper torso of adult shoppers, thereby ignoring shopping carts and small children. Each reconstructed 3D point within the bounds of the VOI is orthographically projected onto the groundplane which has been broken up into square segments corresponding to bins in a histogram. The more 3D points that are projected into a given bin, the higher the bin's count or *occupancy*. To detect people, the most significant peak above a pre-defined threshold is selected from the *occupancy map* and a Gaussian distribution is fit to this peak. If the width of the fitted Gaussian is greater than a second threshold, the Gaussian is determined to belong to a pedestrian and the region belonging to this person is removed from the occupancy map. This process is then reiterated, selecting the next highest peak in the residual occupancy map, until there are no more significant peaks remaining. It should be noted that this type of approach has also been proposed

using *sparse* disparity, for example in [24] where a simple connected components algorithm is employed to cluster points that constitute part of the extracted peak area in the map.

Techniques using this stereo camera setup have the same disadvantages as 2D techniques that employ this viewpoint, see section 2.3.1. In general, they are only applicable to indoor scenarios, which restrict the maximum height at which the camera can be placed. For example, for retail scenarios the ceilings are only 8-9 feet high [166]. This short height can be restrictive as the field of view can be limited unless a wide field of view lenses is employed. However, as previously mentioned, this type of lens can result in significant occlusion problems in all but the central portion of the image [15]. Therefore with overhead camera viewpoints a trade-off exists between the field of view and occlusion. An advantage to using stereo cameras over single cameras is that this trade-off can be removed.

If the 3D groundplane is calibrated with respect to the stereo rig then 3D points can be orthographically projected onto the groundplane no matter what orientation the camera rig is positioned at, therefore allowing the occupancy map approach of [166] to be applied from stereo cameras mounted at more oblique angles. In this manner, the advantages of mounting the camera at an oblique angle, which maximises viewing volume, *and* that of an overhead view, which simplifies person modeling and reduces occlusions, can be exploited, see figure 3.14(d) which illustrates the occupancy map of figure 3.14(c). Features which are obtained from 3D points that are orthographically projected onto the groundplane are known as *plan-view statistics*. The occupancy map of [166] is one such plan-view statistic, others such as *height maps* and *volumetric occupancy maps* are introduced shortly.

Before other plan-view statistics are introduced it should be noted that variations on the occupancy map approach of [166] are presented in the literature that make the approach more robust to more complex scenes from an oblique camera viewpoint. For example, in [166] background objects within the bounds of the VOI can be consistently detected as false-positives if their occupancy count is greater than the required threshold. Some techniques, such as [24] attempt to remove these false-positives by further classification using width, height and motion information of detected objects as inputs to a Time Delay Neural Network (TDNN) classifier. Other approaches, such as that of [15] use background subtraction based on colour and depth models to obtain foreground points, see figure 3.14(c), the 3D position of these foreground points are then used to obtain the occupancy map, see figure 3.14(d). It should also be noted that background *occupancy* models have also been employed in the literature [167]. Pedestrian regions may then be extracted from the foreground

occupancy map in a similar manner to that of [166].

Techniques using occupancy maps tend to fail to segment pedestrians correctly when they are highly occluded. These false-negatives occur due to a significantly lower proportion of a person being visible in the input images which results in a lower number of 3D points and therefore lowers the occupancy count for the region of that pedestrian below the required peak threshold. For similar reasons, occupancy maps cannot detect a person far from the camera because the number of 3D points on a distant person is too small to create a significant peak on the occupancy map [105]. Therefore, it can be seen from these examples, that adding a fixed amount to the occupancy map for each 3D point favours the detection of closer objects. To overcome this, a technique is required to compensate for the depth of a pixel. This could be achieved by scaling the occupancy accumulation value based on range [166], or by normalising the occupancy values along the z-axis [49]. However, this normalisation based on range must be done carefully as a poor normalisation can lead to either no peaks occurring over the required threshold, or the technique becoming overly susceptible to noise.

To overcome these problems another type of plan-view statistic, called a *voxel histogram*, is proposed in [105]. This technique projects variable size 3D voxels instead of 3D points orthographically on the floor plane. For each 3D point, a 3D voxel is created that is scaled according to the point's distance from the camera. The orthographically projected voxel may cover a number of bins on the groundplane histogram. The volumes are then accumulated in a similar manner to occupancy maps. As voxels from near-by pixels tend to overlap, this allows people farther away from the camera to meet the required threshold to be segmented as a person as long as they are not highly occluded. However, in crowded situations the peaks from multiple people often connect, resulting in under-segmentation.

A different approach is taken in [21], which introduces another plan-view statistic called the *height map*. The height map is used to complement two of the occupancy map's failings; namely its lack of virtually all object shape information in the vertical dimension, and the decrease in saliency in the occupancy map when the person is partially occluded by another person or object [21]. The height map is similar to the occupancy map but each groundplane bin contains the highest point above the ground-level plane that is projected into that bin. It is effectively a simple orthographic rendering of the shape of the 3D point cloud when viewed from overhead [21] – see figure 3.14(f).

The detection of objects from the height map can be achieved in a similar manner to that of

occupancy or volumetric maps, for example [139, 168] simply threshold the height map and use connected component analysis to obtain pedestrian regions. However, using this approach the movement of relatively small objects at heights similar to those of peoples heads, such as when a book is placed on an eye-level shelf, can appear similar to the motion of a person in a height map [15]. To overcome this [139, 168] filter the final regions on the basis of their size and height, removing blobs with sizes and heights inconsistent with people. However, these techniques are still affected by noise and problems when people are in very close proximity to one another [168].

To address these issues the authors in [15] apply the use of *both* height and occupancy maps to detect foreground objects that are significant in both sets of maps. Initially, a threshold is applied to the occupancy map to obtain significant object presences – see figures 3.14(d) and (e). This occupancy threshold is set so that half of an average-sized person must be visible to the stereo pair in order to exceed it [21]. The height map is then pruned using the thresholded occupancy map, see figures 3.14(f) and (g). Finally, people are detected if their height is over a given threshold *and* their occupancy is over a threshold. This allows foreground noise and small, non-person foreground objects, which appear to be located at relevant heights, to be ignored. However, depending on the height threshold, children or pedestrians in wheelchairs may not be detected.

All these plan-view statistic based approaches can have difficulties when dealing with the substantial occlusion of a pedestrian, where the occupancy count is unlikely to reach the minimal required thresholds. This problem can actually be seen as a trade-off on the resolution of the groundplane bins; the resolution should be small enough to represent the shapes of people in detail but also must consider the limitations imposed by the noise and resolution properties of the depth measurement system. If the bins are too large then pedestrians may be under-segmented resulting in people becoming merged together into single regions. If the bins are too small then there may be many missed detections as the occupancy maps may never reach the required thresholds or the required peaks may become noisy and sparse. In general, the groundplane is divided into a square grid with a resolution of between 1cm [122] to 4cm [15]. This generally favours reducing under-segmentation and leads to assumptions that substantial occlusion does not occur before a person has been detected and added to a list of tracked objects, which can allow statistical models to be built up that can be then used to segment a pedestrian during occlusion, as seen in section 2.4.2.

Finally, the accuracy of these techniques is also dependent on their adopted pedestrian model. Techniques that do not adopt a model and instead assume that a pedestrian corresponds to just a

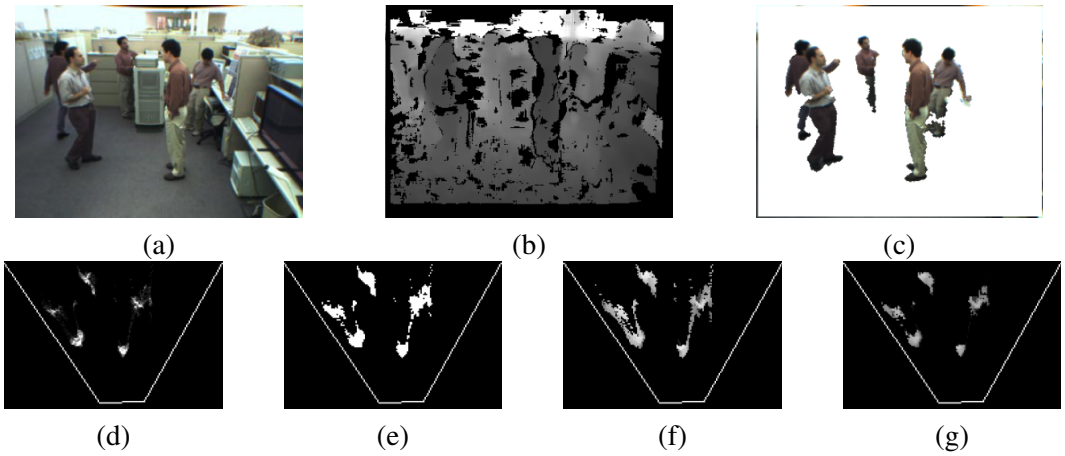


Figure 3.14: From [15]; (a) Example input image; (b) Disparity; (c) Foreground regions obtained using depth and colour models; (d) Occupancy map; (e) Thresholded occupancy map; (f) Height map; (g) Masked height map.

connected region in a thresholded image can result in under-segmentation if pedestrians become too close. Techniques that do adopt models must ensure that they are correct in size; if they are too large then under-segmentation can occur, in addition pedestrians can be missed if a proportion to their occupancy has already been be covered by a previous close-by detection; if they are too small then false-positives may occur if they do not use up their full quota of occupancy, and so may leave an occupancy bin value that exceeds the required threshold on its fringes [21]. A constant size of a model, however, is difficult to define as different people, including children and adults, and different poses, have different associated heights and widths.

In the pedestrian detection technique proposed by the author, a new plan-view statistic is incorporated into the framework. In the proposed technique, the plan-view statistic is used as a guide in the proposed clustering process, and is not subject to fixed thresholds. Thus it is not subject to the threshold problems outlined above. In addition, the proposed approach do not quantise the 3D space into discrete bins, and so a quantisation resolution does not have to be considered. Finally, a basic human biometric model is incorporated which is automatically tailored in size for each pedestrian during their segmentation. As such, the trade-off inherent in model size is eased.

3.5 Pedestrian Tracking using 3D Information

As with 2D pedestrian tracking – see section 2.4 – there are two broad types of methodologies for tracking algorithms that use 3D information; *continuous detect-and-track* and *single detect-and-track*. In this section, neither approach is specifically discussed as in general, the 3D information is

mainly used to resolve ambiguities and therefore can be used within either approach. For example, in a single detect-and-track approach the 3D information can be used to determine a *definitive* depth ordering for appearance models to reduce ambiguity when occlusion occurs. Similarly, for continuous detect-and-track techniques the third dimension can be applied to obtain more robust similarity measures between previous tracks and pedestrians detected in the current frame.

When tracking pedestrians using 3D information, the position of person in the current frame is typically used to disambiguate between pedestrians in subsequent frames. This position is usually defined by some 3D feature point of the object, for example the 3D position of the top, bottom or centroid of the person [164], or the centroid of the person's head region [131, 75]. Some of these features are more robust than others, for example the centroid of a 3D cluster can be more robust than the top or bottom of the cluster region, particularly in outdoor scenes where the object may be far from the camera [41]. However, during occlusion the real and estimated centroids of a full body region may differ significantly, whereas the centroid of the 3D head region can be more robust and remain relatively unaffected.

In general, the most useful feature point that can be obtained for tracking pedestrians is one that lies on the groundplane. This allows tracking of pedestrian positions with respect to the groundplane, which is favourable when compared to that of tracking in 3D with respect to an arbitrary 3D world origin. Tracking using this technique can simplify the prediction of a pedestrian's position in the subsequent frames significantly, for example it is inherently assumed that if a person moves within a scene, then they are moving across a groundplane. Using 3D co-ordinates, however, a pedestrian model is free to move with an extra degree of freedom, allowing hypothesised positions of pedestrians to be illogically above or below a groundplane in subsequent frames. In addition, there is no significant loss of information using this transformation, for example the mapping between 3D to 2D can be a Euclidean transformation, which would allow Euclidean distances between tracks and pedestrians to be calculated.

A groundplane feature point can be obtained using a number of techniques. One of the most basic is to calibrate the groundplane with respect to the stereo camera rig, obtain some 3D feature point, such as a 3D centroid of a region, and orthographically project this point onto the groundplane in a similar manner to that of obtaining plan-view statistics. This is an approach that is adopted in the pedestrian tracking approach proposed in this thesis – see section 6.2.1.1. Another set of approaches include the use of a groundplane homography, or 2D projective transformation, which maps points on the groundplane in one image, to points on the groundplane in a second im-

age – see section 4.3.2. This information can be employed to match groundplane features between images resulting in 3D positions that are on the groundplane. Examples, of this approach can be seen in; (a) [68] which obtains the point of contact of the principal axis of persons in multiple images as the required groundplane point; (b) [63] which uses homography and colour information from multiple cameras to obtain ground locations of people in each view; and (c) [137] in which the orthographic projection of shape-from-silhouette regions and homography information are applied to obtain the centre of object location likelihood maps as the groundplane feature points.

3.5.1 Plan-view Appearance Models

The advantages of plan-view statistics can also be used to add robustness to tracking applications. Using traditional 2D based approaches, most appearance-based tracking methods encounter difficulty in selecting and adapting the appropriate template size for a tracked object, because the size of the object in the image varies with its distance from the camera [15]. However, plan-view representations of people are largely invariant to these changes. In addition, the pedestrian model is simplified when viewed from overhead so the complexity of the spatial models can be reduced somewhat.

The *single detect-and-track* technique adopted in [15] employs the use of height and occupancy maps for appearance features. In the approach, the template for matching corresponds to twice the average torso width of people, a relatively high Kalman gain is used in the update process so that templates adapt quickly, and it is assumed that appearance of the pedestrian in groundplane space varies smoothly over time. In fact, for simplicity, the authors assume that there is *no change* in the plan-view statistics between frames, which is relatively accurate if the frame rate is sufficiently high. For the measurement step, a search of the surrounding neighbourhood of the predicted person position is made and the best match is selected if it is above a certain threshold. In this approach, the correlation is a weighted sum of similarity between occupancy and height maps, distances of position from predicted position and a comparison of Gaussians of *all* previously tracked persons to the Gaussian at this position. This final weight discourages the matching of multiple people to nearly the same plan-view location. In this approach, the potential appearance of new people is detected *after* all known people have been tracked and removed from the occupancy and height maps using techniques described in section 3.4.1.

As with most tracking methods based on adaptive templates, the templates of [15] tend to *drift* off of moving people and onto neighbouring plan-view image regions that are varying less rapidly

over time. These relatively static regions might correspond to a non-moving person, a moved non-person background object or a shadow [16]. In addition, the tracker occasionally swaps the identities of closely interacting people, despite large differences in the colour of their clothing.

To overcome these problems [16] adds plan-view colour templates to the appearance model of a pedestrian. To eliminate drift it applies a *re-centring* scheme that is applied on each frame after tracking has completed. The plan-view colour template used in [16] is that of the pixel that corresponds to the highest point in each bin, see row 3 in figure 3.15. The authors then divide their appearance model into short-term and long-term components. The short-term model makes use of height and colour templates, see figure 3.15. At each time step, correlations between a track and a pedestrian are made by aligning the model centres and obtaining pixel-to-pixel correlations. Then after after tracking is completed, the template models are replaced with the current plan-view image data at locations centred on the estimated person locations. The long-term appearance model describes features that are relatively independent of body pose and activity over the duration of tracking. The body shape is defined as an estimate of the persons standing height, and colour is defined as a histogram quantised coarsely (e.g. 4 bins) in each colour channel. The long-term model is updated is very slowly, equivalent to a time constant of several minutes, to increase its robustness to temporary occlusions or brief posture changes. In the approach, both models are used for matching in a Bayesian classification framework using a solution that obtains the maximum *a posteriori* (MAP) probability, whereby possible match hypotheses are evaluated exhaustively within windows centred at predicted person locations. These windows must be large enough to account for reasonable prediction errors and inter-frame person acceleration [16]. Finally, the effects of template drift are reduced by adopting a re-centring scheme that is applied on each frame after tracking has completed. In this process, each pedestrian centre is repositioned onto the centre-of-mass of the local plan-view occupancy. It is only after this re-centring that the appearance model for the pedestrian is updated.

It should be noted, however, that template drift is generally only an issue with single detect-and-track approaches, and as such the re-centring technique is generally not required in continuous detect-and-track techniques. It should also be noted, that although plan-view colour models are are largely invariant to changes in distance [15], with the use of 3D stereo techniques other, more traditional, can be scaled appropriately. Therefore the reasons to use plan-view colour models are minimised, as it may eliminate distinguishing colour information that may occur *below* the highest point in each bin. For example, in row 3 of figure 3.15 notice how there is little green colour

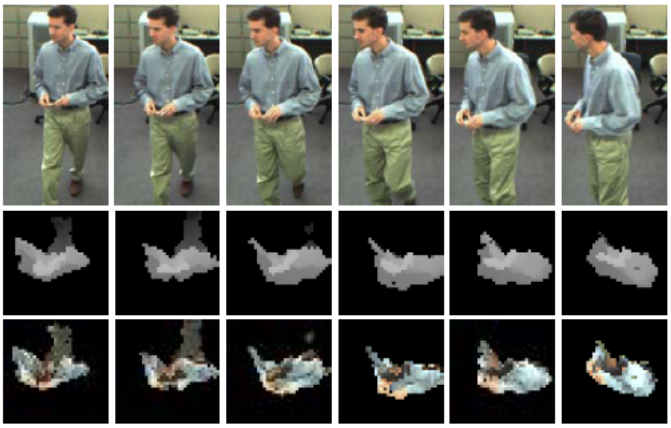


Figure 3.15: Templates for tracked person from [16]; (Row 1) Colour input; (Row 2) Height templates; (Row 3) Colour templates.

which corresponds to the colour of the persons trousers in row 1 of the same figure. In subsequent frames, this lost colour information may be the key to correctly determining the continuation of the person's track in the presence of several ambiguous tracking matches. In the proposed pedestrian tracking approach a continuous detect-and-track methodology is applied. Therefore, for these reasons plan-view appearance models are not applied for tracking in this work.

3.6 Summary

In this chapter the main categories of approaches for pedestrian detection and tracking using 3D information were outlined. These techniques apply 3D information in a variety of approaches to; increase robustness to illumination changes for background modelling; reason and segment objects during occlusion; reduce false-positives using the size, position and density of objects in Euclidean co-ordinates; and, improve tracking results.

However, many of these techniques still present a trade-off in terms of under-segmentation when pedestrians become close together, and detection of false-positives. This trade-off is inherent in the single pre-defined human model that is chosen to represent the entire human class, and in the case of plan-view statistics, the resolution of the quantisation space for the groundplane. In these cases, if the model (or groundplane resolution) is too large then under-segmentation occurs, but if it is too small then false-positives due to over-segmentation and noise can occur.

A major flaw in many of the pedestrian detection techniques described in this chapter is that few (if any) attempt to obtain the highest *quality* disparity map that is possible within reasonable time constraints given the scene features and temporal information. Instead, the techniques tend

Figure 3.16: Disparity map after background subtraction (from [14]).

to apply standard stereo correspondence algorithms, and often the reasons for a specific choice of algorithm are not justified. The pedestrian detection techniques then tailor their processing based on the perceived quality of the resultant disparity map. For example, the technique described in [14] (see section 3.4 and figure 3.13(c)) applies the standard disparity estimation technique from the Triclops stereo vision library that is packaged with the Digiclops [169] stereo vision camera system manufactured by Point Grey Research [170]. This technique, however, does not obtain disparity from within homogeneous regions, leaving holes of missing disparity within foreground and background objects. After background subtraction, the disparity map can become even more sparse and noisy, see figure 3.16. The technique applied in [14] then groups these foreground blobs together using a set of heuristics. This grouping stage may be eliminated from the proposed technique, however, if a robust and dense disparity estimation technique was employed. The same disparity estimation technique from the Triclops stereo vision library is applied by a number of plan-view statistic based approaches [16, 15, 21], see figure 3.14(c). Similar, non-dense, techniques are also employed in [74, 24, 122].

In the following chapter an overview of disparity estimation techniques is given and the first contribution of this thesis is presented. A dynamic programming based stereo correspondence technique is defined that has been specifically developed for applications involving pedestrian detection. The technique reduces artifacts in the calculated disparity map via a number of novel enhancements to the dense disparity estimation algorithm including a *dynamic* disparity limit constraint, which limits the region where a match for a pixel in one image can occur in a second image, and the use of highly reliable matched pixels, known as Ground Control Points (GCPs), to help guide results. This disparity estimation technique is quantitatively evaluated against other standard techniques using a synthetic dataset designed to mimic typical pedestrianised scenarios and difficulties.

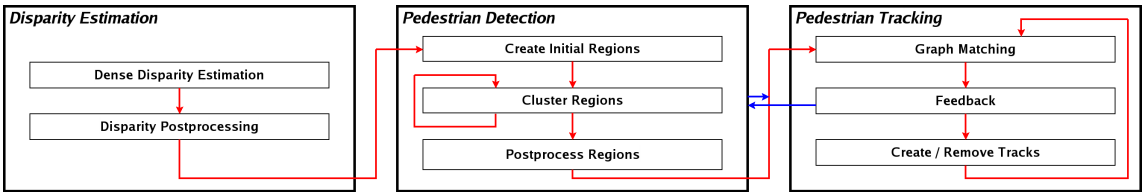


Figure 3.17: High-level System Overview.

The generation of a robust disparity map can be seen as part of the first module in the pedestrian detection and tracking system of this thesis, illustrated in figure 3.17. This disparity map is then post-processed to remove artifacts and constrain the 3D points to a volume of interest, whereupon it is used as an input to the pedestrian detection module – discussed in chapter 5 – which is the second major contribution in this thesis. In this approach the post-processed disparities are clustered together into pedestrian regions via an iterative region growing framework that incorporates 3D information, groundplane estimation, non-quantised plan-view statistics and a basic human biometric model that is automatically tailored for each pedestrian during their segmentation. Finally, the clustered regions are post-processed to remove background regions and noise.

The final module and contribution of this work is that of pedestrian tracking using a continuous detect-and-track methodology, using a weighted bipartite graph, is described in chapter 6. A maximum-weighted maximum-cardinality matching scheme is then employed, with additional kinematic constraints and explicit occlusion analysis, to obtain the best match from previous tracks to currently detected pedestrians. A number of separate rollback loops are used to backtrack the pedestrian detection module to various states to further reduce over-/under-segmentation of detected pedestrians and increase tracking robustness.

CHAPTER 4

Disparity Estimation

4.1 Introduction

In the previous chapter it was shown that if two image points, u_1 and u_2, are projections of the same real-world point X, then the 3D position of this point can be obtained via triangulation (see section 3.2.2). The difficulty with this approach is that, in general, given u_1, the corresponding point u_2 is *not* known *a priori*. Therefore, before X can be obtained as a point in 3D space a search must be undertaken to find the correct match, or corresponding point, to u_1. This is known as the *correspondence problem* and is the most difficult stage of the stereo reconstruction process. Techniques to solve the correspondence problem are known as *stereo matching* (or *disparity estimation* or *correspondence*) algorithms and their development constitutes one of the most active research areas in computer vision. Robust disparity estimation, however, is difficult to achieve, especially in areas of homogeneous colour or occlusion.

In this chapter, a brief overview of stereo correspondence techniques is presented in section 4.2, which outlines the main approaches proposed in the literature along with their respective strengths and weaknesses. In section 4.3, the first major contribution of this work is presented. This contribution outlines a stereo correspondence technique, designed specifically for pedestrian surveillance type applications, which incorporates a number of novel enhancements to increase the robustness of the final disparity map. Finally, in section 4.4 the proposed technique is quantitatively and visually evaluated against both synthetic and real-world datasets. The results of this evaluation illustrate the robustness of the proposed approach compared to a variety of state of the art disparity estimation algorithms.

4.2 Stereo Correspondence Matching

In order to recreate the 3D structure of a scene using stereo vision techniques, the correspondence problem between image points must be resolved. There are a huge variety of approaches proposed in the literature. In general, the choice of which technique to use depends on the application and scene characteristics.

As an output, most stereo correspondence methods compute a uni-valued disparity map with respect to a reference image, which could be one of the input images, or a reconstructed cyclopean image positioned between the two images [171]. Each point in this disparity map, $d(x, y)$, encodes the correspondence point for each pixel (x, y) in the reference image. In [172] the ideal disparity map is described as smooth and detailed; continuous and even surfaces should produce a region of smooth disparity values with sharp and accurate boundaries, while small surface elements should remain and not be eliminated. However, it is not easy for a stereo algorithm to satisfy these two requirements. Algorithms that can produce a smooth disparity map tend to miss the details and those that can produce a detailed map tend to be noisy [172].

At the highest level, stereo correspondence matching approaches can be split into two groups, namely *passive* and *non-passive*. In passive techniques, no attempt is made to contaminate the scene with extra information to make the correspondence problem easier. This is not the case in non-passive techniques. For example, in [173] the approach presented relies on using a pair of cameras and one or more light projectors that cast structured light patterns onto the scene. Each camera uses the structured light sequence to determine a unique code (label) for each pixel. Finding inter-image correspondence then trivially consists of finding the pixel in the corresponding image that has the same unique code.

In addition, stereo correspondence algorithms can be split into two further sub-groups, namely *dense* and *sparse* disparity techniques. For dense disparity maps, the depth for *all* pixels in an image is calculated. The resultant dense disparity maps, however, can include very noisy, unreliable, depth values, especially in non-textured and occluded areas in the scene. The alternative to this is to apply a sparse disparity technique, where depth values in non-reliable homogeneous regions are generally removed or not calculated. However, sparse and dense disparity algorithms share many similarities, and in general a dense disparity map can be generated from a sparse disparity, using for example *region growing* or *seed-and-grow* techniques such as [174, 175]. In these techniques, sparse disparity point matches are propagated to less textured pixels in an iterative manner, usu-

ally in a best-first strategy. However, in the literature review presented here, only dense passive techniques are considered.

To give a precise summary of every known stereo correspondence algorithm would be a mammoth task, so the literature review proceeds by breaking up stereo correspondence algorithms into their constituent stages. Each stage is then introducing separately, describing the main techniques typically applied for each stage. Most stereo correspondence algorithms generally perform (subsets of) the following four steps [171]:

1. Matching cost computation
2. Cost aggregation
3. Disparity computation / optimisation
4. Disparity refinement

The actual sequence of steps taken depends on the specific algorithm. Before each of these stages is described, a number of constraints commonly used to simplify the correspondence problem are outlined.

4.2.1 Constraints

As stereo matching is a difficult process, especially in areas of homogeneous colour with little texture, many algorithms make use of one or more constraints to simplify the correspondence problem. These constraints can be spit up into three subgroups [176, 12];

1. **Geometric constraints imposed by the imaging system**
    - The **Epipolar Constraint** states that the correspondence problem be narrowed from a 2D image search to a 1D line search using geometric constraints (see section 3.2.2.1).
    - The **Disparity Limit Constraint** states that a correspondence has to occur within a certain disparity range along the epipolar line, if a match is not found within this disparity range then no match exists. This disparity limit should be large enough to obtain the correct disparity of the closest object to the camera.
    - The **Ordering Constraint** states that corresponding feature points typically lie in the same order on the epipolar line. This constraint is violated if a small narrow object is much closer to the camera than a background object (see figure 4.1(a)) however in

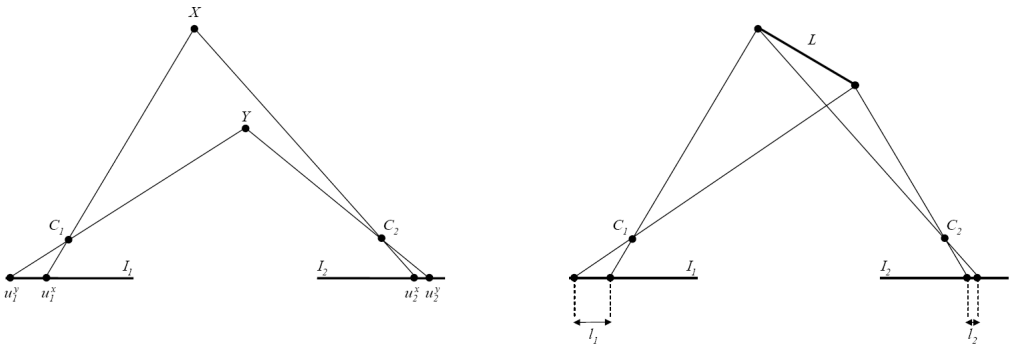
many application scenarios, such as pedestrian surveillance, this case is unlikely to occur.

2. **Geometric constraints arising from the objects being looked at**

   - The **Geometric Similarity Constraint** states that geometric characteristics of the features found in the first image (e.g. length or orientation of a line segment, region or colour) do not differ much from the corresponding features in the second image. However, slanted surfaces violate this constraint if the baseline between the two cameras is large [149] (see figure 4.1(b)). Notice how the orientation of the line L is such that it is closer to fronto-parallel in camera C_1 than in camera C_2, thus the length of the projection of the line segment is greater in I_1 than in I_2 – this effect is called *foreshortening*.
   - The **Disparity Smoothness Constraint** states that the disparity changes slowly almost everywhere in an object, except at depth discontinuities that occur at object boundaries.
   - The **Figural Disparity Constraint** states that if a point in I_1 lies on an image edge, then the corresponding point in I_1 should also lie on an edge, as well as adhering the Disparity Smoothness Constraint. However, this assumption may be violated if the camera baseline is large.
   - The **Uniqueness Constraint** states that a pixel in the first image can only correspond to *one* point in the second image and vice versa.
   - The **Mutual Correspondence Constraint** states that if u_1 is matched to u_2 via a search for a corresponding point to u_1 in I_2, then if the search is reversed and a match for u_2 in I_1 is undertaken, then u_2 must be matched to u_1. If this constraint is violated then the match is not reliable and should be ruled out. This bidirectional matching technique is a typical method to deal with occlusion [172]. Inconsistent matches, which create holes in the disparity map, can be post-processed and filled in using interpolation techniques.

3. **Physical constraints**

   - The **Photometric Compatibility Constraint** states that the colour intensities of a point in the two images are likely to differ only a little. However, due to differing angles of the camera this may not be possible without a pre-processing step. This dif-



(a) (b)

Figure 4.1: (a) Ordering constraint violation; (b) Foreshortening.

ficulty is due to differing camera gain properties, differing surface normals can also produce large errors when dealing with specular reflections, etc.

4.2.2 Matching Cost Computation

Stereo algorithms differ according to the type of features they attempt to match and to the type of features they use to represent them [176]. These features include single pixels, edges and a variety of image regions. In order to match these features, a matching cost function is required.

4.2.2.1 Intensity Based Correlation

Many dense disparity methods apply *correlation based* cost functions. For these approaches the Photometric Compatibility Constraint is enforced, meaning an assumption is made that the corresponding pixels between images have very similarly intensities [12]. One of the most common, and basic, matching costs is the *Sum of Squared intensity Differences* (SSD), used in [177, 178, 179], where the square of intensity differences between two corresponding points is obtained for each colour channel. These values are then summed together. Therefore, the lower the SSD correlation score, the better the match between two candidate pixels. Another approach is the *sum of absolute intensity differences* (SAD), as used in [180, 181, 182]. This function performs better than the sum of squared differences in the presence of outliers.

Another traditional matching cost is *normalised cross-correlation*, which behaves similar to SSD [171]. Used by many techniques [183, 18, 182, 175, 184], it is less sensitive to noise and illumination transformations between the input images, such as contrast and brightness modifications, at the expense of being more computationally expensive.

4.2.2.2 Non-Parametric Correlation

The problem with parametric measures is that they are highly reliant on the Photometric Compatibility Constraint – see section 4.2.1. However, discrepancies between camera parameters, such gain or bias [185], may lead to a violation of this constraint whereby the colour intensities of corresponding points in a stereo image set differ significantly. If this case occurs, then the accuracy of disparity estimation techniques that are based on parametric correlation measures is likely to decrease. However, non-parametric measures, such as rank and census transforms, are insensitive to these camera parameters. Non-parametric local transforms use ordering of local intensity values, not the values themselves. Two such examples are the *rank* and *census* transforms [185]. In the rank transform, the number of pixels in the local region whose intensity is less than the intensity of the centre pixel is calculated. To compute correspondence, the sum of absolute values of differences on the rank-transformed images [24] is minimised. The census transform maps the local neighbourhood surrounding a pixel u to a bit string representing the set of neighbouring pixels whose intensity is less than that of u. Two pixels of census transformed images are compared for similarity using the Hamming distance, i.e. the number of bits that differ in the two bit strings. To compute correspondence, the Hamming distance is minimised [186].

There exist a variety of gradient-based measures, which are insensitive to differences in camera gain or bias, that uses edges, [187, 188], corners [189, 188] or the direction of the intensity gradients [190]. However, the use of these features is generally only possible for sparse disparity techniques.

4.2.3 Aggregation of Cost

In general, binocular stereo matching is ambiguous as there are often multiple equally good matches if the matching quality is evaluated independently at each point purely by using image properties [178]. In order to increase reliability, some disparity estimation techniques aggregate the matching cost by summing or averaging over a support region [171]. The correlation between matching pixels is then made over a similar support region in each image using some matching cost function. These methods can be defined as *area*, *block* or *window* based methods. The manner in which this support from the local neighbourhood is calculated varies between algorithms and is related to fundamental assumptions the algorithms make about the scene and its surfaces [178].

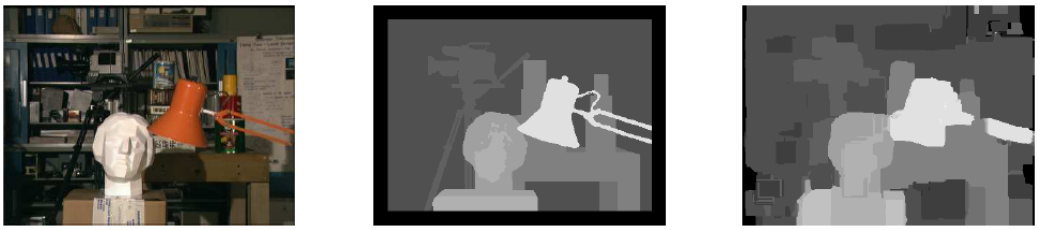
The simplest technique is to use square windows of fixed size. For example [181] uses 19×19

pixel windows. Typically only one disparity is found per block and each pixel contained within the block has the same disparity, i.e. the Disparity Smoothness Constraint is inherently enforced. However, since these local aggregation methods assume a constant disparity a trade-off is introduced. The probability of a mismatch goes down as the size of the correlation window and the amount of texture increases [191]. However, using large windows leads to a loss of accuracy, due to each pixel in the window having the same disparity. This loss of accuracy results in missing important image features, especially at disparity discontinuities – see figure 4.2 taken from [17] and notice how the disparity "spreads out" in (c) at depth discontinuities when compared to (b). This erroneous effect is known as a *corona*. Therefore, a trade off exists between matching accuracy and loss of information. Deciding on a static window size is therefore difficult and arbitrary window sizes are common.

Techniques have been proposed to try and overcome this trade off inherent in fixed window sizes. A multiple, or shiftable, window method is applied in [192] whereby the cost function is aggregated over a number of predefined windows, which are located at different positions to the currently tested pixel – see figure 4.3 taken from [18] which uses 9 different support windows. The window with the lowest correlation value is then retained as the best match. Although this technique improves performance, the shape of the chosen support window may be inappropriate for the pixels near arbitrarily shaped depth discontinuities.

A different approach is presented in [193], which uses a fixed window but aggregates support for a pixel with a Gaussian weighted support. Therefore, points further away have gradually less influence. This correlation technique is presented in an iterated framework, whereby each iteration increases the effective size of the aggregation area. To prevent boundary blurring, the diffusion process can be stopped at any iteration when a "clear" minimum amongst all the candidate disparities arises. Therefore, the diffusion is only iterated at ambiguous matches.

A similar approach is taken in [178] which propose the use of *adaptive windows*, which are square windows that extend by different amounts in each of the four directions. This extension is implemented in an iterative framework, whereby a model of disparity uncertainty in the current aggregation window, determined from the intensity and disparity variances of points from the window centre, is used to search for a window with the least uncertainty for each pixel in the image. The algorithm iteratively updates the disparity estimate for each point by choosing the size and shape of a window until it converges. However this technique is highly dependent on the initial disparity estimation and is computationally expensive. Moreover, the shape of a support window



(a) (b) (c)

Figure 4.2: Corona effect [17]: (a) Input image; (b) Groundtruth disparity; (c) Corona effect.

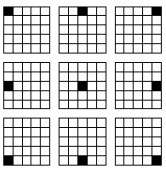


Figure 4.3: Multiple windows [18]. The black dot in each window represents the tested pixel.

is constrained to a rectangle, which also may not be appropriate for the pixels near arbitrarily shaped depth discontinuities [17].

Finally, some techniques attempt to overcome the trade-off between large and small windows by applying hierarchical coarse-to-fine frameworks. These techniques are iterative algorithms, where results from coarser levels are used to constrain a more local search at finer levels [171]. Two such approaches for local based techniques are proposed in [177, 194]. The first approach, *fine-to-fine*, initially obtains disparity using large matching windows, the window size is then reduced to refine disparity estimates using the previous window disparities as guidance. The second approach, *course-to-fine*, is computationally more efficient by using a fixed window size but is implemented at several levels of image resolution computed by sub-sampling Gaussian smoothed images.

4.2.4 Disparity Computation and Optimisation

Stereo algorithms that produce dense disparity maps can be further classified as *local* or *global* based on the type of optimisation method used [195]. Other disparity estimation techniques, such as dynamic programming, exist that are located in between the two main techniques, whereby each pair of matching epipolar lines, or scanlines, are treated independently, but within each scanline the algorithm finds a minimum solution to a global scanline cost function.

4.2.4.1 Local optimisation

For local optimisation techniques, the disparity value at each pixel is chosen independently of other pixels [195]. Normally these methods perform a local Winner-Takes-All (WTA) optimisation scheme at each pixel or window. The fundamental problem of these techniques is that each pixel is treated in isolation. Therefore, they do not take into account the fact that a match at one point restricts others due to global constraints resulting from stereo geometry and scene consistency [172]. In local optimisation techniques, the emphasis is on the matching cost computation and on the cost aggregation steps. Computing the final disparities is trivial: simply choose at each pixel the disparity associated with the minimum cost value [171]. However, due to the lack of global consistency in disparity estimation techniques of this class, local optimisation methodologies are not adopted for the proposed disparity estimation approach in this thesis.

4.2.4.2 Global optimisation

As opposed to local-based techniques, disparity estimation approaches applying global optimisation seek a solution that minimises some cost function for *all* pixels in the input images. Global optimisation methods attempt to find this solution by formulating the problem in an energy-minimisation framework, which usually contains a data term and a smoothness term [195]

$$E(x, y, d(x, y)) = E_{match}(x, y, d(x, y)) + E_{smooth}(x, y, d(x, y))$$

where $d(x, y)$ is the disparity for the point (x, y), $E_{match}(x, y, d(x, y))$ is a correlation function, and $E_{smooth}(x, y, d(x, y))$ is a smoothness function that usually defines the correlation of $d(x, y)$ with some subset of neighbouring pixels. This smoothness term is set to ensure that the disparity changes slowly across the image, therefore many global optimisation techniques tend to often skip the aggregation step [171], however this is not always the case. For example [195], applies an aggregation scheme that allows windows have up to two disparity values, one for a foreground object, and one for a background object.

The exact implementation of this energy function is dependent on the choice of correlation function, the required smoothness assumptions and the optimisation function applied. Once the global energy has been defined, a variety of algorithms can be used to find the solution that provides the best global set of correlation matches. Any efficient optimisation algorithm must use two techniques to find this solution: *exploration*, to investigate new and unknown areas in the search

space, and *exploitation*, to make use of knowledge found at points previously visited to help find better points [196]. These two requirements are contradictory, and a good search algorithm must find a trade-off between the two. For global based optimisation, the most popular optimisation techniques include simulated annealing, mean field annealing, graph cuts and belief propagation.

An experimental comparison of simulated annealing, graph cuts and belief propagation methods is presented in [197], whereas graph cuts and two belief propagation techniques are compared in [198]. In both cases the graph cuts algorithm has the best performance in terms of disparity estimation. In addition, in terms of efficiency the graph-cuts technique has an advantage over a number of other global techniques. For example, it is shown in [171] to be more efficient than simulated annealing. In [198] it also performs well, but not as efficiently as the *accelerated* belief propagation algorithm. However, in general global based techniques are memory intensive and computationally expensive, especially if the disparity search range is large [199]. For example, [171] records times of 10–30 minutes for disparity estimation using graph cuts and simulated annealing on standard disparity datasets. For these reasons, a disparity estimation technique based on global optimisation is not adopted in this thesis.

4.2.4.3 Dynamic Programming

There exists a hybrid of local and global techniques whereby each pair of matching epipolar lines, or scanlines, are treated independently, but within each scanline the algorithm finds a minimum solution to a global scanline cost function. While the 2D-optimisation of the global energy function can be shown to be NP-hard for common classes of smoothness functions, dynamic programming techniques can find the global minimum for independent scanlines in polynomial time [171]. As with global based techniques, a global function for each scanline is created. Finding the correct disparities is akin to finding the path in this space that takes the cheapest route through the cost values [200].

In general, dynamic programming based approaches explicitly enforce the *ordering constraint*, which requires that the relative ordering of pixels on a scanline remain the same between the two views. However, this is not always the case, for example the *Scanline Optimisation* (SO) approach of [171] does not utilise visibility or ordering constraints. Instead, a disparity is assigned at each point such that the overall cost along the scanline is minimised. In this case, if the global cost function's smoothness term is set to zero, this would be equivalent to a Winner-Takes-All (WTA) optimisation.

(a) (b)

Figure 4.4: Dynamic programming [19]; (a) Input image; (b) Streaking.

In general, however, the ordering constraint is enforced which results in a reduction of the overall search space. As with global optimisation techniques, a variety of cost functions have been proposed incorporating differing smoothness costs. For example, [19] proposes a dynamic programming approach whereby the cost of a match sequence is made up by a function of a constant penalty for each occlusion, a constant reward for each match, and a sum of the dissimilarities between the matched pixels. Additional techniques have also been proposed to increase the robustness of dynamic programming approaches. One such technique is to incorporate highly-reliable matches termed *Ground Control Points* (GCPs) [18, 182] into the matching process. In [18] these GCPs are found by matching edge features. The cost for the disparity of the GCPs is then set to zero, whereas the cost for all other disparities for these GCPs is set very high. This forces the dynamic programming algorithm to incorporate as many GCPs as possible into the final solution. These techniques can be used to significantly increase robustness of the final disparity map while simultaneously reducing the computational complexity of the algorithm.

The main issue associated with dynamic programming techniques is that global constraints are not efficiently enforced between individual scanlines. This typically leads to streaking effects [201] – see figure 4.4 taken from [19], notice the streaking effects in (b). In order to enforce inter-scanline consistencies, [182] introduces a two-pass dynamic programming technique that performs optimisation both along and across the scanlines. The first pass optimises the solution along scanlines, the second pass then uses the subsequent results to optimise the solution across scanlines to provide a final disparity. This technique does, however, come at the cost of increased computational expense, due to the optimisation process being iterated twice. In the proposed disparity estimation technique in this thesis – as presented in the following section – a single pass dynamic programming based technique is employed that enforces inter-scanline consistency via the use of GCPs, a dynamic disparity constraint and a novel cost function.

4.2.5 Refinement of Disparities

Most stereo correspondence algorithms compute the disparity of a given point to be a discrete value between 0 to n, where n is defined by the disparity limit constraint. For a number of applications such quantised disparity maps can lead to poor results, e.g. for view synthesis applications. To remedy this situation, many algorithms apply a sub-pixel refinement stage after the initial discrete correspondence stage. Other common post-processing techniques include removing ambiguous pixels using bi-directional matching, smoothing the resultant disparity map or filling in holes in the disparity map via interpolation.

4.3 Proposed Disparity Estimation Technique

The first major contribution of this thesis outlines a robust dense disparity estimation technique that has been specifically designed for pedestrian detection and tracking based applications. The technique employs an assumption, which is also made in other parts of this work, that a relatively flat groundplane is present in the scene and that this groundplane can be pre-calibrated with respect to the stereo camera rig. This groundplane is generally present in most scenarios from which detection of pedestrians is desired, such as as crossroads, streets, airports, railway stations, etc.

The disparity estimation technique is based on a dynamic programming based stereo correspondence approach that includes a smoothness cost in *both* the vertical and horizontal directions, but unlike that of [182] only a single pass of the algorithm is required. In addition, the technique reduces artifacts in the calculated disparity map via a number of enhancements to the dense disparity estimation algorithm. An overview of the algorithm is given in figure 4.5.

The first enhancement involves a 2D projective transformation of the input rectified images into *groundplane space*. This technique aligns the images horizontally with respect to a groundplane. The result of this transformation is that corresponding points on the groundplane in separate images transform to the same point in a reference image in groundplane space, whereas points above or below the ground plane would not transform consistently. Once in groundplane space, the resultant disparity search space can be reduced, as disparities that correspond to 3D points positioned *under* the groundplane are not searched for and, thus the possibility of false matches and the computational complexity of the dynamic programming algorithm can be reduced.

The second enhancement involves the use of highly reliable matched pixels, known as Ground Control Points (GCPs) [18, 182], to help guide results. The use of GCPs in the dynamic program-

ming stage of the algorithmic process is similar to that of [18, 182], whereby the GCPs are used to help anchor the final dense disparity map to correct disparity values. The proposed technique used to obtain GCPs differs greatly from other techniques. It can be seen as a four stage process, see figure 4.5; (1) using the input images in groundplane space, *Foreground Activity Regions* (FARs) are determined in which good GCPs are likely to be found; (2) GCP disparities are determined from within FARs using a number of metrics to ensure that the GCPs are highly reliable matches; (3) these GCPs are interpolated throughout the image by applying predefined knowledge of the orientation of the 3D position of the GCPs, with respect to the 3D groundplane within the scene. The resultant GCPs are clustered together to form *GCP Regions*; (4) background GCPs are found using background disparity and edge models.

The third enhancement introduces a technique for obtaining a *dynamic* disparity limit constraint to further improve GCP selection and dense disparity generation, in addition to reducing algorithmic complexity. As outlined in section 4.2.1, a disparity limit constraint can be used to limit the possibility of where a match for a pixel can occur, thereby making algorithms less susceptible to incorrect matches and reducing computational expense. An issue for many algorithms is selecting an appropriate value for the disparity limit. It should be large enough allow the correct disparity of the closest object to the camera to be obtained, but small enough to allow some saving on computational expense. For a given set of input images, the optimal *a priori* setting of this value is an ill-posed problem as the distance to the closest object to the camera is not known until its disparity has been obtained. In the proposed approach a *dynamic* disparity limit constraint is obtained, whereby the disparity limit is determined during the disparity estimation process for each separate scanline. Two separate schemes for obtaining the disparity limit are employed; the first technique uses FARs to limit the disparity search for GCPs; the second technique uses the resultant GCPs to limit the disparity search during dynamic programming.

The final enhancement applies a novel scanline cost function for the dynamic programming algorithm that enforces inter-scanline consistency, while simultaneously providing a framework for allowing pixels to be set to the disparity of the groundplane at a reduced cost. This final feature is employed to increase the likelihood of the disparity of ambiguous pixels being set to that of the groundplane disparity so that they can have no influence, in either computational expense or algorithmically, upon the subsequent pedestrian detection technique.

An overview of the disparity algorithm is given in figure 4.5, in the following sections each of the main stages is examined in more detail. In section 4.4, a comparison of the proposed technique

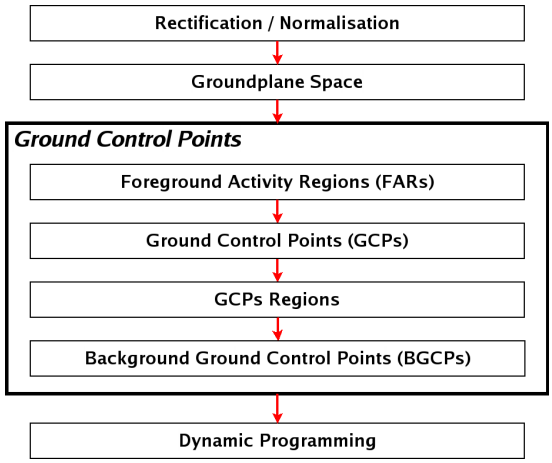


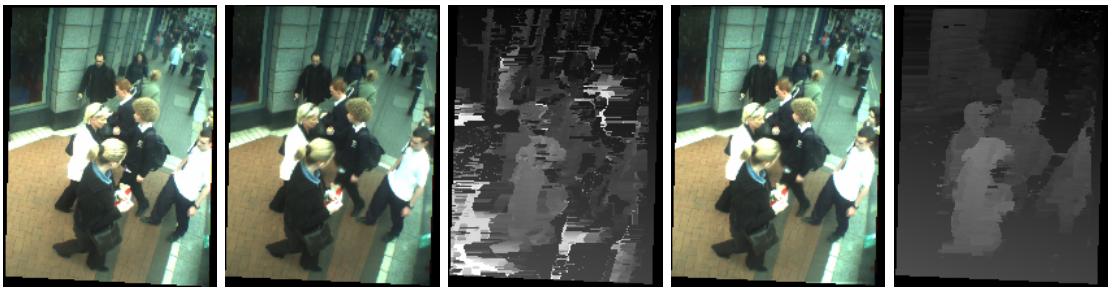
Figure 4.5: Disparity estimation overview.

with a number of local and dynamic programming disparity estimation techniques is presented on both synthetic data and real-world indoor and outdoor scenarios encompassing varying lighting and camera positions.

4.3.1 Rectification and Normalisation

The first stage of the algorithmic process is to ensure that the input images are both rectified and normalised. Image rectification is a pre-processing step employed by most short-baseline disparity estimation techniques that re-samples the input images so that epipolar lines are parallel within images. The technique and advantages of rectification are presented in section 3.2.2.2.

A second pre-processing step normalises the colour intensities of the input images, which reduces problems that occur in real-world camera stereo rigs, such as differences in camera gains or reducing the effects of specular reflections that can be projected onto the cameras. These differences can cause a violation in the Photometric Compatibility Constraint (see section 4.2.1) as the colour intensities of corresponding points in a stereo image set can differ significantly. To counter these issues, the images are normalised by ensuring the average values for each of the red, green and blue channels are the same for each input image. For each set of input stereo images, the brightest single image is determined as the one with the largest value of summed red, green and blue colour channels for all pixels within that image. All pixels in each of the other input images are then normalised by the required amount so that the average values of the channels are equal. A possible disadvantage of this technique is that the input stereo images are assumed to have 100% overlap with each other, and as this is not the case an object that can be viewed in only one of the images can cause a deviation from the true normalisation values. However, due to the



(a) (b) (c) (d) (e)

Figure 4.6: (a) Input image I_1; (b) Non-normalised input image I_2; (c) Non-normalised disparity map; (d) Normalised input image I_2; (e) Normalised disparity map.

small baseline of the stereo camera employed in our experiments the image overlap is very large with respect to non-overlapping regions and as such this scenario has little practical effect on the normalisation technique.

The effects of normalisation on the proposed disparity estimation technique is illustrated in figure 4.6[1]. In figure 4.6(c) the input images have not been normalised and the resultant disparity map is both noisy and erroneous, especially on the groundplane, which appears significantly brighter in I_1 than in I_2. Many on these erroneous disparity values have been eliminated in figure 4.6(e), where the *only* change to the overall algorithm is that the input images have been normalised (notice how the overall brightness in I_2 between figures 4.6(b) and (d) has increased). The difference in colour intensities between corresponding pixels of I_1 and I_2 is therefore less significant and allows more robust disparity estimation using intensity based correlation metrics.

4.3.2 Groundplane Space

Rectification aligns the images vertically, so that epipolar lines are parallel. A *second* pre-processing step is then applied that aligns the images horizontally with respect to the groundplane. This technique is based on the application of a groundplane homography [165].

Homography An homography, H, also referred to as a *projective transformation* or *collineation*, is an invertible mapping within 2D projective space, $\mathbb{P}^2$, such that three homogeneous points u_1, u_2 and u_3 lie on the same line if and only if Hu_1, Hu_2 and Hu_3 are collinear [148]. A 3×3 homography matrix can be used to describe both the relationships between real-world planes and a camera's image plane, see figure 4.7(a), and between two perspective views of a planar object, see figure 4.7(b).

[1]The disparity estimation technique used in figure 4.6 is the one described in the following sections.

Assuming a point, X, on a planar surface, π, is projected to the point u_1 in the image plane I_1, an homography, $H_{1\pi}$, exists from π to I_1 – as shown in figure 4.7(a). This homography can be viewed as a mapping between real-world groundplane points and the position in I_1 where these points are projected. If a second camera observes the same scene and the same planar surface, π, then a second distinct homography, $H_{2\pi}$, exists from π to I_2. However, as these two homographies have a common reference plane π, the composition of the two homographies, $H_{\pi 1}$ and $H_{\pi 2}$, results in a third homography [202], H_{12}, which exists from I_2 to I_1:

$$u_1 \cong H_{1\pi} H_{2\pi}^{-1} u_2 \tag{4.1}$$

$$u_1 \cong H_{12} u_2 \tag{4.2}$$

where $\cong$ denotes equality up to a scale factor due to the use of homogeneous coordinates.

This homography from I_2 to I_1, illustrated in figure 4.7(b), is known as a *plane induced* homography as it depends on the position of the real-world plane π. The result of this projective transformation is that if u_1 and u_2 are both projections of the real-world point X, then $u_1 \cong H_{12} u_2$ *iff* X is on the plane π, whereas the farther the point X is off the plane the greater the disparity between $H_{12} u_2$ and u_2's actual corresponding point in I_1.

In this work, a *groundplane* induced homography is obtained between the input images with respect to the 3D groundplane within the scene. In order to obtain an homography, in general, four corresponding groundplane points between images I_1 and I_2 must be obtained. However, as the input images have been rectified the homography can be represented as a shear and translation of I_2 to I_1 across a single image axis (in this case the x-axis). The resultant homography therefore takes the form

$$H_{12} \cong \begin{vmatrix} \alpha & -\beta & \gamma \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} \tag{4.3}$$

and only three corresponding groundplane points between images I_1 and I_2 must be obtained to constrain α, β and γ. More information on how to obtain these values is presented in appendix A.

Groundplane Space Using a *groundplane* induced homography, a projective transformation of each point in the input image I_2 can be made via equation 4.2. The subsequent images, I_1

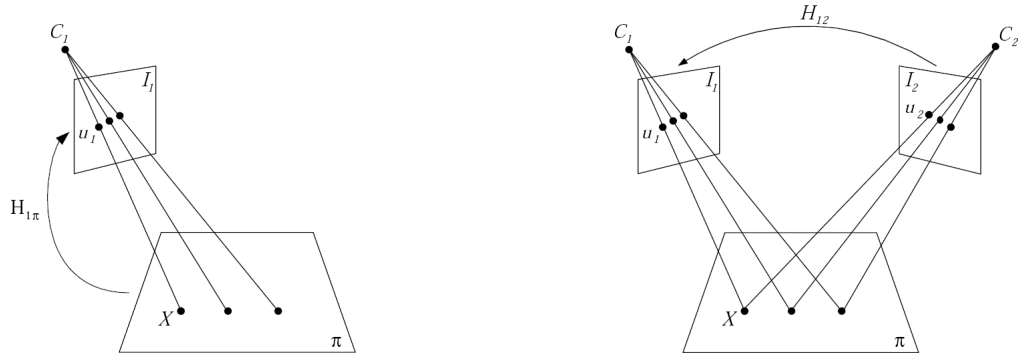


Figure 4.7: (a) Homography $H_{1\pi}$ between a world plane and the image plane I_1; (b) Homography H_{12} induced by a plane.

and $H_{12}I_2$ are then used as the input to the proposed disparity estimation technique. Due to the groundplane induced homography transformation, the resultant input images I_1 and $H_{12}I_2$ are referred to as being in *groundplane space*.

When the input images are in groundplane space, the projection of a real-world 3D groundplane point, X, onto the image planes of I_1 and $H_{12}I_2$ occurs at the *same* 2D co-ordinate, whereas the points corresponding to real-world 3D points located above or below the groundplane are not projected to the same co-ordinate, see figure 4.8(d). In groundplane space, it can be seen that the further above the groundplane a 3D point is, the greater the disparity between points within the input images, see figure 4.8(h) which is the section within the red boundary of figure 4.8(e). In this image, the red dotted line is overlaid onto the edge of the building, which is perpendicular to the groundplane, in I_1, and the blue dotted line corresponds to the same edge in $H_{12}I_2$. Notice how the two lines intersect at the groundplane but diverge further apart with increased height above the groundplane.

These properties of groundplane space are used for two enhancements to the disparity estimation technique. Firstly, as assumption can be made that no 3D points appear below the groundplane in the scene, therefore the disparity estimation can be undertaken in groundplane space, rather than using the rectified images. This is advantageous as the number of disparities to be searched for a given scanline can be reduced, resulting in eliminating possible mis-matches and increasing computational efficiency as the search space for corresponding points is trimmed. Secondly, the properties of groundplane space can be can be applied to find areas, called Foreground Activity Regions (FARs), in which depth discontinuities are likely to occur. These regions are typically a good place to extract a large number of strong GCPs as they tend to occur in areas of high texture.

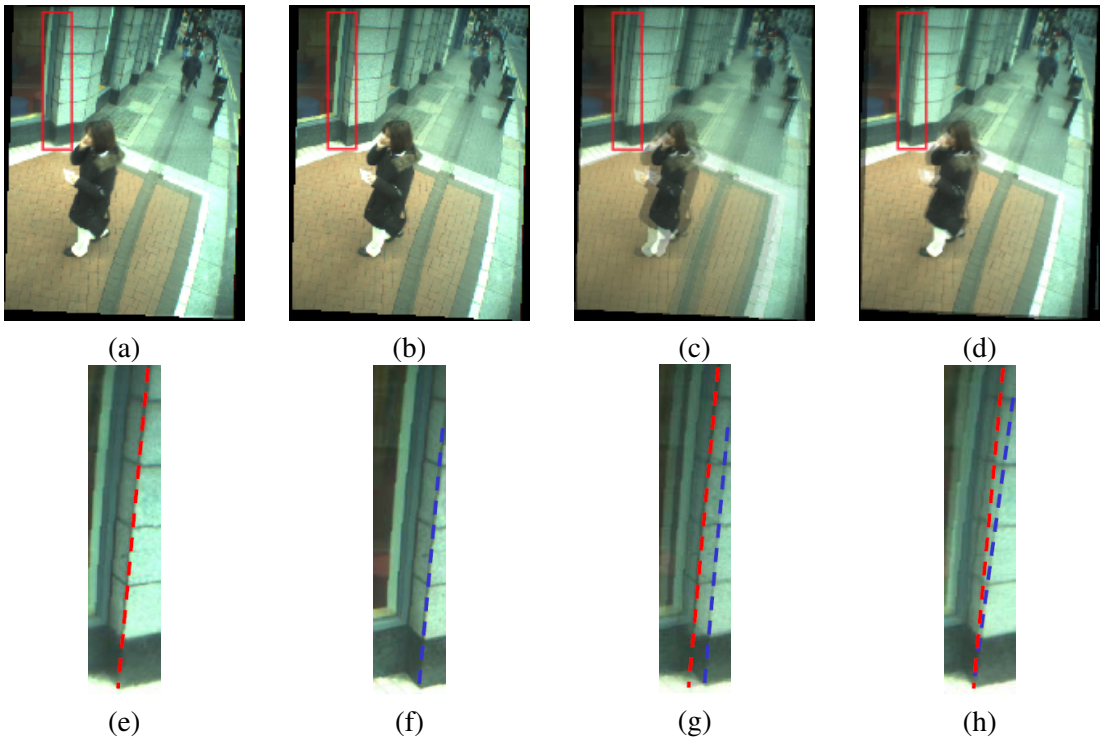


Figure 4.8: Groundplane space (a) I_1; (b) I_2; (c) I_1 and I_2 overlaid; (d) I_1 and $H_{12}I_2$ overlaid; (e) Zoomed section of I_1; (f) Zoomed section of I_2; (g) Zoomed section of I_1 and I_2 overlaid; (h) Zoomed section of I_1 and $H_{12}I_2$ overlaid.

4.3.3 Foreground Activity Regions (FARs)

Ground Control Points (GCPs)[2] have been found to help guide stereo correspondence techniques resulting in more accurate results. However, false GCPs can severely degrade the final matching results [182]. Introducing stricter constraints on the selection of GCPs could decrease the number of false GCPs, but could also reduce the total number of GCPs, leading to the possibility of an insufficient number of GCPs being available to successfully guide the matching process.

The properties of the groundplane space can be applied to find areas, called Foreground Activity Regions (FARs), in which depth discontinuities are likely to occur. These regions are typically a good place to extract a large number of strong GCPs as they tend to occur in areas of high texture. FARs occur due to foreground objects being above the groundplane, and therefore their corresponding pixels are not overlaid in groundplane space, see figure 4.9(a). For an illustrative example, consider figures 4.9(c) and (d), which depict the areas of I_1 and $H_{12}I_2$ that are contained within the red bounding box to the left of the person's midsection in figure 4.9(a).

As the white region in the two image sections is part of the pedestrian's hand, which is fore-

[2]It should be noted that within the notation "Ground Control Points (GCPs)", the word "Ground" does not refer to the groundplane. Rather, within the context of this thesis, GCPs are defined to be highly reliable pixel matches that, due to the technique used to obtain them, tend to be located above the groundplane.

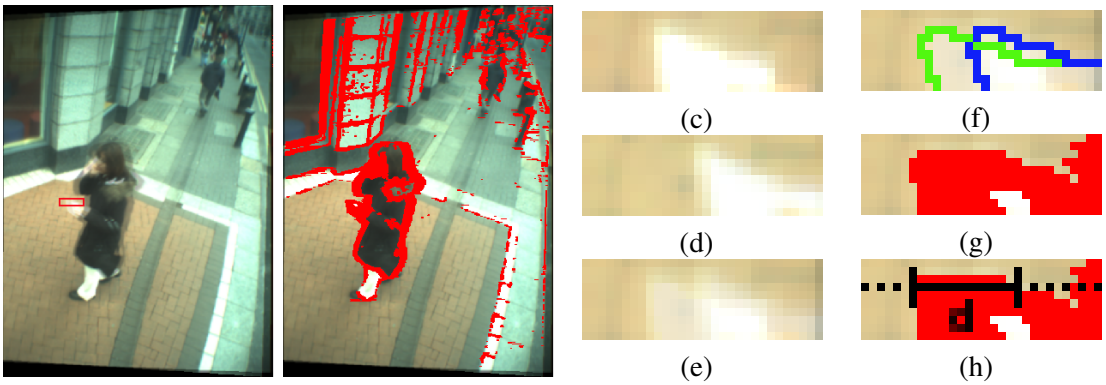


Figure 4.9: (a) Groundplane space; (b) FARs; (c) I_1 section; (d) $H_{12}I_2$ section; (e) I_1 and $H_{12}I_2$ sections overlaid; (f) Matching edges; (g) FAR section; (h) Dynamic disparity.

ground and above the groundplane, a given co-ordinate, (x, y), which is on the hand in I_1 does not correspond to the same real-world point as the point (x, y) in $H_{12}I_2$ – see figure 4.9(e). This can be more clearly seen by viewing the position of the edge points in I_1 (in green), with respect to the corresponding edge points in $H_{12}I_2$ (in blue) in figure 4.9(f). The key property is that, in groundplane space, neither the two sets of edges nor *any* pixel between the two edges match in both colour intensity and gradient information. Such pixels are denoted in red in figure 4.9(g). Let these regions of non-correspondence be called FARs. An important aspect is that, in general, if there is a jump in disparity, then this disparity discontinuity is incorporated within a FAR.

Once FARs are determined, the FAR is then searched for matching GCPs between images I_1 and $H_{12}I_2$. However, an advantage to obtaining FARs to determine GCPs, instead of just regions or points of high texture within an image, is that a *dynamic* disparity limit constraint can be determined for each GCP match. If a *static* disparity limit is employed, the value of this limit should be ideally set to the disparity of the closest object to the camera rig. However, this is an ill-posed problem as the distance of the object to the camera is not determined until the object's depth is obtained. The constraint is therefore usually fixed at the largest expected disparity that should occur in the scene. Using the FARs regions, it is possible to roughly estimate the *maximum possible* disparity. In general, if there is a jump in disparity within a FAR then a match for every point in the FAR is at a disparity less than the width of that FAR. In figure 4.9(h) notice how the width of the FAR, d, is greater than that of the disparity of the two edges. The search for GCPs in each FAR is implemented using this variable value, d, as the maximum disparity limit constraint.

In order to obtain a FAR, each pixel in groundplane space is examined. For a given pixel, (x, y), let $rgb_1^{(x,y)} = (r_1^{(x,y)}, g_1^{(x,y)}, b_1^{(x,y)})$ be the red, green and blue channels in I_1 respectively,

and let $rgb_2^{(x,y)} = (r_2^{(x,y)}, g_2^{(x,y)}, b_2^{(x,y)})$ be the corresponding values for $H_{12}I_2$. In addition, let $grad_1^{(x,y)} = \{grx_1^{(x,y)}, gry_1^{(x,y)}, ggx_1^{(x,y)}, ggy_1^{(x,y)}, ggx_1^{(x,y)}, ggy_1^{(x,y)}\}$ be the vector of gradients in I_1 for (x, y), where for the gradients for the red channel are defined as

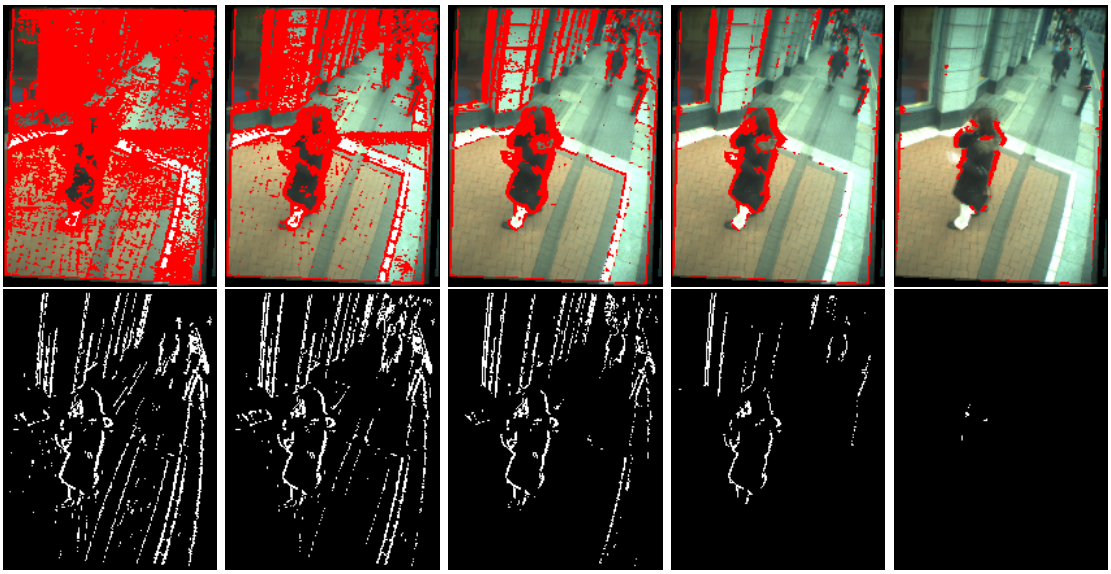
$$grx_1^{(x,y)} = r_1^{(x+1,y)} - r_1^{(x-1,y)} \tag{4.4}$$

$$gry_1^{(x,y)} = r_1^{(x,y-1)} - r_1^{(x,y+1)} \tag{4.5}$$

and similar gradient values exist for the green and blue channels in I_1 and all channels in $H_{12}I_2$. The pixel (x, y) is then defined as part of a FAR if

$$\begin{aligned} SSD(rgb_1^{(x,y)}, rgb_2^{(x,y)}) &> 3 \times (t_{FARs}^{rgb})^2 \text{ , or} \\ SSD(grad_1^{(x,y)}, grad_2^{(x,y)}) &> 6 \times (t_{FARs}^{grad})^2 \end{aligned}$$

where SSD is the sum of squared differences and the two thresholds t_{FARs}^{rgb} and t_{FARs}^{grad} are user defined. In these inequalities, the thresholds represent the maximum difference per channel expected. Therefore, as SSD is applied the thresholds are squared and multiplied by the number of channels in the input vectors. The setting of these thresholds can have a large effect on the number of GCPs obtained, see figure 4.10. If the values are too low, then a high number of FARs are not created and therefore are less likely to fully span a depth discontinuity, leading to artificially low disparity limits and therefore matches for GCPs may not be found, see figure 4.10(e). If the values are too high, see figure 4.10(a), then the disparity limits can become artificially high. This may lead to more GCPs, but it also increases the possibility of incorrect GCP matches occurring. However, the technique used to obtain GCPs is very robust at removing ambiguous GCPs, so increasing the thresholds setting has little effect on the creation of false-positive GCPs. However, due to the addition of more FARs the scene, it becomes less likely that Background GCPs (see section 4.3.6) are able to propagate. In this work, the thresholds are generally set at $t_{FARs}^{rgb} = 20$ and $t_{FARs}^{grad} = 10$, which gives a good balance between the number and quality of GCPs obtained and the creation of reliable FARs.



(a) (b) (c) (d) (e)

Figure 4.10: FAR user defined parameters, (top row) FARs, (bottom row) Positions of obtained GCPs; (a) $t^{rgb}_{FARs} = 5$, $t^{grad}_{FARs} = 2.5$; (b) $t^{rgb}_{FARs} = 10$, $t^{grad}_{FARs} = 5$; (c) $t^{rgb}_{FARs} = 20$, $t^{grad}_{FARs} = 10$; (d) $t^{rgb}_{FARs} = 40$, $t^{grad}_{FARs} = 20$; (e) $t^{rgb}_{FARs} = 100$, $t^{grad}_{FARs} = 50$.

4.3.4 Ground Control Points (GCPs)

Finding GCPs within FARs is a two stage process; (1) Initial GCP (IGCP) matches are found using dynamic disparity limits obtained from FARs, these IGCPs are filtered using metrics obtained on the scanline to remove ambiguous matches; (2) the IGCPs are post-processed, to remove possible incorrect matches, using information from multiple scanlines. After this stage, all remaining IGCPs are set to be GCPs and the resultant GCPs are extended horizontally across scanlines.

4.3.4.1 Initial Ground Control Points (IGCPs)

The first step in this process involves finding Initial Ground Control Points (IGCPs). In general, it is more difficult to determine the accurate disparity of a point that has homogeneous rather than heterogeneous neighbours. For this reason, it is ensured that the matches for the IGCPs are instantiated by edges that have a strong gradient that is vertically oriented with respect to the scanline, i.e. $gry_1^{(x,y)}$, $ggy_1^{(x,y)}$, or $gby_1^{(x,y)}$ must be greater than a threshold, $min_{grad} = 10$. If a pixel meets this requirement, it is labelled as a maximum vertical edge, e [3].

To determine the IGCPs for a given FAR on a particular scanline, the set of all e in each image within the FAR is found. Each e within the FAR of I_1, namely $e1$, is compared to each e in $H_{12}I_2$,

[3] If two neighbours on a given scanline both meet these criteria, the neighbour with the strongest gradient is defined as a maximum vertical edge, e.

namely $e2$, that is within a disparity of the width of the FAR. The two candidate matches are compared using the sum of squared differences (SAD) in their RGB colour intensities *and* their gradient values. If

$$\begin{aligned} SSD(rgb_1^{e1}, rgb_2^{e2}) &\leq 3 \times (t_{GCPs}^{rgb})^2 \text{ , and} \\ SSD(grad_1^{e1}, grad_2^{e2}) &\leq 6 \times (t_{GCPs}^{grad})^2 \end{aligned}$$

where the two thresholds t_{GCPs}^{rgb} and t_{GCPs}^{grad} are user defined, then a possible match between the two edges exists. For an IGCP instantiated at $e1$, with a disparity of $|e1 - e2|$, this match must be the *only* match that satisfies the above constraint, i.e. if $e1$ can be matched to more than one edge in $H_{12}I_2$, then $e1$ cannot be an IGCP. In addition to this constraint, the IGCP match must also be bi-directional, i.e. if the process was reversed and $e2$ was matched to edges in I_1, then $e2$ *must* be matched to only $e1$ for the bi-directional constraint to be held true. These two constraints remove a large amount of ambiguous matches from the IGCP selection process, and due to this technique it becomes apparent why a small value for thresholds t_{FARs}^{rgb} and t_{FARs}^{grad}, as used when creating FARs, does not create more false-positives. Setting these thresholds at small values increases the size of the dynamic disparity limit and therefore increases the likelihood of a false match for the IGCP, but any IGCP that is matched to two or more possibilities is removed unreservedly.

4.3.4.2 Post-processing IGCPs

Finally, to further reduce the likelihood of a spurious match, any proposed IGCPs that have no immediate inter-scanline neighbours that are also marked as IGCPs of similar disparity are excluded.

Determining GCPs based on horizontal matches alone, as in the creation of IGCPs, can still result in false matches occurring, mainly due to noise or if the FAR regions are too small with respect to a disparity discontinuity. The second stage in obtaining robust GCPs clusters IGCPs vertically across scanlines, enforcing inter-scanline consistency between the final choice of GCPs. For a given IGCP in I_1 it is checked to see if there exists one or more IGCPs in the previous scanline within a distance of 1 pixel that also has an absolute disparity difference of 1 pixel. If more than one possible IGCP exists, the IGCP that has the closest colour intensity and gradient information to the current IGCP is chosen as a match and the two IGCPs are chained together. This process is reiterated for each IGCP in each scanline in I_1. All IGCPs that have a chain less than some threshold, set to 3 in our experiments, are removed as possible noise or mismatches.

A final post-processing stage is then employed, which removes IGCPs that do not strictly adhere to the ordering constraint (see section 4.2.1), which states that corresponding feature points typically lie in the same order on the epipolar line. In this approach, a dynamic programming technique is applied to enforce the ordering constraint in ICGPs, where the cost value for the dynamic programming algorithm is set to the amount of IGCPs aligned correctly. This technique therefore maximises the total number of IGCPs that adhere to the ordering constraint, removing possible outliers.

At this stage, each remaining IGCP is declared as unambiguous and set as final GCP. The final stage of this process is to extend each GCP horizontally across the scanline, within the area of the GCP edge where the gradient is greater than the minimum threshold, $min_{grad} = 10$, and is within the FAR. For each GCP, the pixel to the left of the edge in I_1, namely $u1$, and $H_{12}I_2$, namely $u2$ is obtained. If

$$\begin{aligned} SSD(rgb_1^{u1}, rgb_2^{u2}) &\leq 3 \times (t_{GCPs}^{rgb})^2 \text{ , and} \\ SSD(grad_1^{u1}, grad_2^{u2}) &\leq 6 \times (t_{GCPs}^{grad})^2 \end{aligned}$$

then the pixels $u1$ and $u2$ can be seen to corroborate the choice of GCPs, and so $u1$ is set to a GCP with the same disparity. The pixels to the left of $u1$ and $u2$ are then selected, and as long as both are within a FAR and are not homogeneous in colour with its neighbours the process is reiterated. This technique is then re-iterated from the original GCP in the alternative direction.

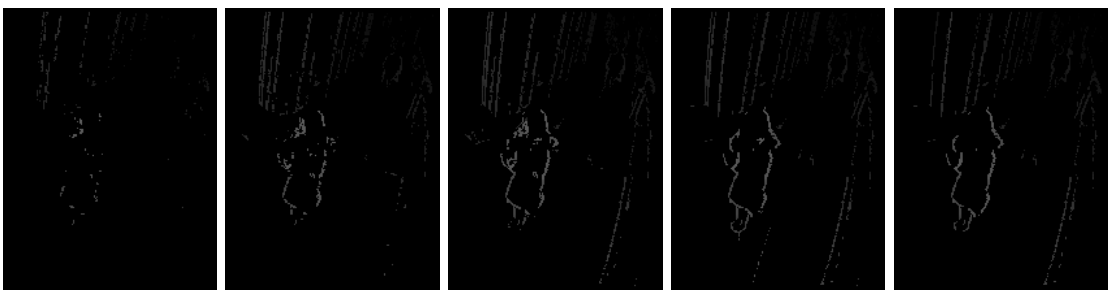
Figure 4.11 illustrates the advantages of using GCPs for disparity estimation in a dynamic programming framework. In this approach, $t_{FARs}^{rgb} = 20$, $t_{FARs}^{grad} = 10$, $t_{GCPs}^{rgb} = 40$, $t_{GCPs}^{grad} = 20$, $t_{DP}^{rgb} = \infty$, $t_{DP}^{grad} = \infty$ and the vertical and horizontal smoothness costs for the dynamic programming approach, which is described in section 4.3.7, are set to 0 and 250 respectively. In this approach, the horizontal smoothness cost is artificially high and causes an over-smoothing in the horizontal direction of the resultant disparity maps, see row 3 in figure 4.11. However, using GCPs can anchor the disparity map to the correct disparities resulting in a large increase in the accuracy of the disparity map, even in the presence of an artificially high smoothing cost.

In general, the smoothness cost is not this high, however setting an appropriate smoothness cost can be difficult for pedestrian detection applications as, in general, pedestrians can appear at a number of depths, and therefore at a number of scales in an input image. Due to these scale differences, if the smoothness cost is set to a high value, pedestrians close to the camera

appear smooth and correctly detailed. However, pedestrians at greater distances, which constitute smaller image regions, can become over-smoothed. This is due to the high cost of changing disparity several times within a small image region. This can result in the DP algorithm choosing an alternative path that keeps the disparity at a constant level, resulting in disparities such as those in figure 4.11 for pedestrians further from the camera, especially if several disparity discontinuities exist between pedestrian regions. However, if the smoothness cost is set to a low value, then the pedestrians further away from the camera may have the correct disparity. There is again a trade-off however, as the closer a pedestrian comes to the camera, the noisier the disparity can become as the pedestrian region is wide and jumps in disparity are cheap to make. Therefore small intensity differences between correctly corresponding pixels can cause large changes in the path the dynamic programming technique takes though the disparity search space. However, in the proposed approach a high number of GCPs can be obtained, meaning that the smoothness costs can be set to a reasonably high level, but yet ensures that the resultant disparity is not overly smoothed as it can be anchored to the correct disparities.

In the proposed approach, the values of the user defined thresholds, t^{rgb}_{GCPs} and t^{grad}_{GCPs}, should be set to a value that is low enough to remove any definite mis-matches between GCPs in colour and gradient, but high enough to allow GCPs to form, see figure 4.12 (where $t^{rgb}_{FARs} = 20$ and $t^{grad}_{FARs} = 10$). In figure 4.12(a), the thresholds are set to low values and therefore result in a small number of GCPs. As the value of the thresholds increase more GCPs are created up to a peak value (see figure 4.12(c)), whereby increasing the thresholds any further results in an overall reduction in the number of GCPs (see figure 4.12(d)). This drop-off in the number of GCPs occurs as higher thresholds allow an increasing number of IGCPs in I_1 to be matched to more than one point in $H_{12}I_2$, and therefore they can no longer be considered as GCPs. Increasing the thresholds to infinity, see figure 4.12(e), results in GCPs being created if there is only 1 possible edge match present in $H_{12}I_2$ for an edge in I_1. However, if the FAR region is not wide enough to cater for the correct depth discontinuity at that point then this GCP is incorrect. For this reason, the best choices for the thresholds are somewhere near the number of peak GCPs (however this peak may shift slightly depending on the input image and choice of FAR thresholds). In our experiments, the thresholds are empirically set at $t^{rgb}_{GCPs} = 40$ and $t^{grad}_{GCPs} = 20$.

Figure 4.11: The advantages of using GCPs; (Row 1) Input images; (Row 2) GCPs; (Row 3) Disparity with a high horizontal smoothing cost without GCPs; (Row 4) Disparity with a high horizontal smoothing cost *with* GCPs.



(a) (b) (c) (d) (e)

Figure 4.12: GCP user defined parameters (Note: $t^{rgb}_{FARs} = 20$ and $t^{grad}_{FARs} = 10$); (a) $t^{rgb}_{GCPs} = 10$, $t^{grad}_{GCPs} = 5$; (b) $t^{rgb}_{GCPs} = 20$, $t^{grad}_{GCPs} = 10$; (c) $t^{rgb}_{GCPs} = 40$, $t^{grad}_{GCPs} = 20$; (d) $t^{rgb}_{GCPs} = 100$, $t^{grad}_{GCPs} = 100$; (e) $t^{rgb}_{GCPs} = \infty$, $t^{grad}_{GCPs} = \infty$.

4.3.5 Ground Control Point Regions

Even with the use of a large number of reliable GCPs, disparity estimation techniques can still suffer from errors in large areas of homogeneous colour. Figure 4.13 illustrates an example scenario. In Figure 4.13(c) the disparity through the midsection of a pedestrian on the left-hand side is incorrect. This type of artifact occurs as there is no texture within the area of the pedestrian's torso, and as such the cost for matching pixels at a number of disparities is almost equal. As no GCPs can be created within this area to guide results, the dynamic programming technique therefore chooses the path that does not include a disparity discontinuity, as the smoothness cost incurred by such a jump in disparity pushes the total cost of this path above other paths that do not include the disparity jump. In the proposed dynamic programming based technique, this type of disparity artifact is removed using two techniques.

The first technique is a vertical smoothness cost, described in section 4.3.7. This vertical smoothness cost is employed in a similar manner to horizontal smoothness costs by enforcing inter-scan line consistency throughout a disparity image. However, the vertical smoothness cost must be relatively low (normally it is set to half the horizontal smoothness cost) as otherwise the resultant disparity map becomes blocky and results deteriorate (as shown in section 4.3.7). Using a vertical smoothness cost, the effect of this type of scenario can be reduced, or in many cases eliminated. However, the lower this cost value, the more likely it becomes that the full artifact will not be removed, especially from objects at a larger distance to the camera.

The second technique applied to help remove this type of artifact uses a similar methodology to a vertical smoothness cost, in that the disparity from other scanlines are used to guide the disparity estimation results. However, unlike the vertical smoothness cost that is incorporated into the dynamic programming algorithm, this second technique applies the use of previously defined GCPs. In this approach, predefined knowledge of the 3D position and orientation of the GCPs with respect to the 3D groundplane within the scene is applied. By applying the reasonable assumption that all the objects in the scene (i.e. people, buildings, etc) are positioned vertically with respect to the 3D groundplane, the resultant disparity map can be improved by extending the GCPs matches across regions of homogeneous texture.

The creation of GCP regions can be viewed as a three stage process; (1) initial GCP regions are created using connected components and the triangulation of GCP disparity points; (2) each point within a GCP region is orthographically projected towards the groundplane, creating addi-

tional GCPs and perhaps causing the merging of separate GCP regions; (3) gaps are filled within each GCP region by horizontally extending the GCPs, however no further clustering of regions is permitted.

The first stage in this process is clustering GCPs into discrete regions. This is achieved using a simple implementation of 8 neighbourhood connected components algorithm [12], whereby two separate GCPs are clustered together if they are neighbouring pixels in I_1 and they have a Euclidean distance in 3D space of less than a threshold, α. The 3D position of a GCP point, $p^{3d} = \{x, y, z\}$, is obtained via triangulation (as defined in section 3.2.2.3). In our experiments, α is set to 25cm so as not to cluster GCPs from two or more distinct foreground objects as it is not expected that GCPs from two distinct objects to be within this close proximity.

The second stage orthographically projects each GCP onto the groundplane and interpolates GCP disparities along this projection path. The 3D position of a GCP point, $p^{3d} = \{x, y, z\}$, can be orthographically projected via the following technique. Let $\{A, B, C, D\}$ represent the equation of the groundplane in 3D, where A, B, and C are the X, Y, and Z components of the groundplane surface normal, and D is the perpendicular distance from the World Euclidean Origin (see section 3.2.1) to the plane. The orthographic projection of p^{3d} onto a groundplane point, q^{3d}, is obtained by finding the intersection of the line defined by the points $\{x, y, z\}$ and $\{x+A, y+B, z+C\}$, and the 3D groundplane. q^{3d} is defined as $\{x+(A*t), y+(B*t), z+(C*t)\}$, where t is defined as

$$t = -\frac{A * x + B * y + C * z + D}{\sqrt{A^2 + B^2 + C^2}} \tag{4.6}$$

Note that the value of t is equal to the Euclidean height of the point p^{3d} above the groundplane.

For a given GCP, each pixel between the two end points p^{2d} and q^{2d} are traversed, where p^{2d} and q^{2d} are the respective projections of p^{3d} and q^{3d} onto I_1. For each point between p^{2d} and q^{2d} the disparity is interpolated and a *new* GCP is created. The path traversal is stopped at any point, (x, y), from p^{2d} onwards if

1. (x, y) is the last edge on the path
2. (x, y) is a GCP
3. $SSD(rgb_1^{(x,y)}, rgb_2^{(x+d,y)}) > 3 \times (t_{GCPs}^{rgb})^2$, where d is the interpolated disparity at (x, y)
4. $SSD(grad_1^{(x,y)}, grad_2^{(x+d,y)}) > 6 \times (t_{GCPs}^{grad})^2$

The third and fourth tests ensure that these extended disparity values do not result in poor correspondence matches. If the second condition is the one to cause the traversal of the path from p^{2d} and q^{2d} to stop, then it is tested to see if the two neighbouring GCPs on the path have a Euclidean distance in 3D space of less than α to each other. If this is the case they are clustered together into the same GCP region. Figure 4.13(d) shows the results from this stage.

The final stage in this process involves traversing the image horizontally with respect to the groundplane and interpolating GCPs across a single GCP region. For each GCP, the 3D path that is horizontal with respect to the groundplane is obtained. This 3D path is then projected onto the image plane. This path is traversed from left to right and, if possible, GCP disparities are interpolated across areas of homogeneity between GCPs of the same GCP region as long as the tests defined in items 2, 3 and 4 of the previous list are satisfied for every point within the gap. Figure 4.13(e) shows the results from this final stage. This addition to the disparity estimation technique has the potential to greatly improve the resultant disparity map in certain scenarios. For example, figure 4.13(f) shows the dense disparity map obtained with this process. Notice how the disparity flows more smoothly vertically and the consistency of the disparity within homogeneous regions is improved.

These extended GCP regions are handled slightly differently in the dynamic programming algorithm as they are *interpolated* values and as such should be used to guide results less stringently than the previously defined GCPs. In addition, care must be taken to to ensure that incorrect GCPs (that can be extended vertically and horizontally also) are minimised. This can be achieved by setting the values of t^{rgb}_{FARs}, t^{grad}_{FARs}, t^{rgb}_{GCPs} and t^{grad}_{GCPs} to appropriate values as previously discussed. For the remaining sections in this chapter, GCPs that are obtained via interpolation are referred to as *GCP regions*, whereas previously defined GCPs are simply called GCPs.

4.3.6 Background Ground Control Points (BGCPs)

As a final stage in obtaining GCPs, the use of temporal data is applied to extract GCPs that correspond to background regions within the scene. By making the assumption that the background can be relatively well modelled, using background suppression techniques described in section 2.2.1, then this information can be applied to obtain pixels that appear to be caused by background regions in the current scene. These pixels can be set as background GCPs (BGCPs), which can be used in a similar manner to GCP regions in the dense disparity estimation approach. The first stage in this process is to create two models of the background; a background *gradient* model, and

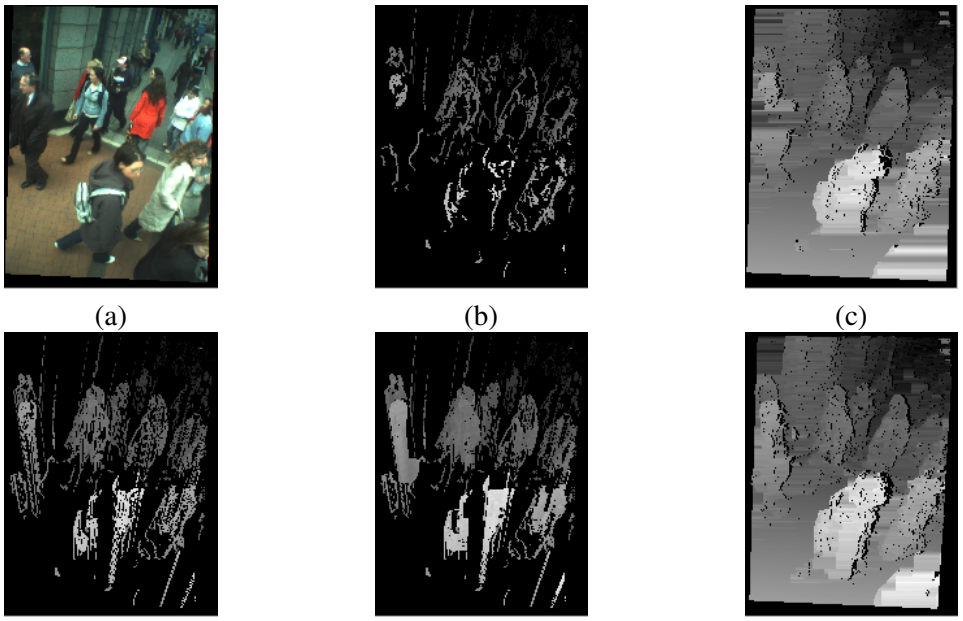


Figure 4.13: (a) Input image; (b) GCPs; (c) Original disparity; (d) Extend GCPs vertically; (e) Extend GCPs horizontally; (f) Improved disparity.

a background *disparity* model. The background gradient model was investigated as image gradient features are more robust to rapidly changing lighting conditions, which can occur in real-world environments.

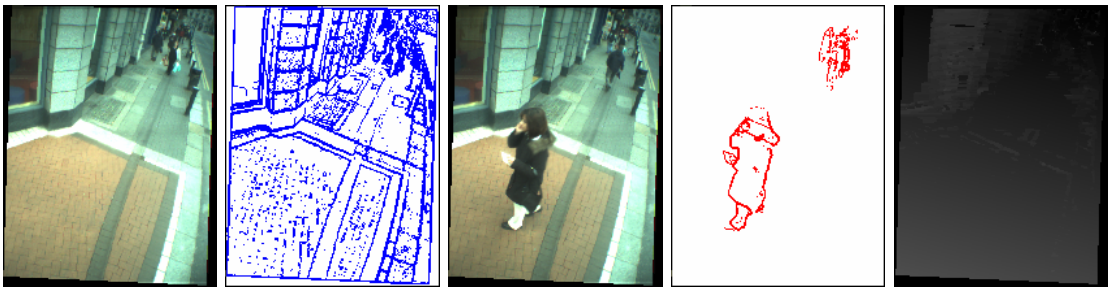
It should be noted however, that unlike most applications that employ background suppression techniques, the accuracy of the end result here is *not* application critical. In this work, the two background models (of which the background gradient model is applied again in the pedestrian detection module), are used to *guide* the final results, and as such highly accurate foreground segmentation is neither required nor sought. To illustrate why these models do not have to be highly accurate, consider the two cases in which they are applied in this work. The first case is to obtain BGCPs, which are used to guide the dense disparity estimation technique. If, however, the background models fails at specific times then the BGCPs are not extracted, however the dynamic programming algorithm can still proceed and obtain dense disparity without the use of BGCPs. If however, a number of BGCPs are obtained, then they may be applied to help guide and improve results from the stereo estimation algorithm. The second case involves using the background gradient model to eliminate background *objects* during the post-processing stage of the pedestrian detection module. The removal of the final objects is based on the background and foreground gradients within the *whole* object. Therefore, even if incorrect classification of some pixels within the object occurs, the correct classification of an object as foreground or background can still occur

as long as the ratio between good to bad classification remains above a threshold that is presented in chapter 5. Each background model is now presented, followed by the technique applied to obtain BGCPs.

4.3.6.1 Background Gradient Model

A background gradient model based on a uni-modal distribution is employed whereby a separate Gaussian distribution is fitted to each of the 6 gradient values. In order to cope with changing illumination and background conditions a two-layered framework [203, 67], with a small learning rate parameter is created. Layered modeling allows multiple levels of background models to be maintained within the background subtraction framework [203]. In the proposed approach, the first layer represents the usual background model, whereas the second layer can be thought of as a cache, which is used as a second, short-term, background model. In this approach, the background model is updated with a very slow learning parameter but only when the current pixel is declared as background. Alternatively, if the pixel is declared as foreground then it is stored in the cache for n frames. In addition, if the current pixel differs significantly from what is stored in the cache then the value in the cache is reset. In our experiments, the background model is created over a small number of frames, some of which consist of foreground objects, with a relatively short update time. After this training period, the value of n is set to a large value of 500 frames. However, in our experiments, many of the sequences are less than 500 frames. Therefore, for these sequences the background model is effectively unadaptive to any sudden background or global illumination changes. However, as will be seen in section 6.3.3, the system remains robust to rapidly changing illumination conditions due to the application of the background models within the pedestrian detection and tracking framework.

In this model, a pixel is declared as foreground if the value of any of the current images 6 gradient values is outside m standard deviations from the mean. In our experiments, the standard deviation is initially set to $m = 4$ or 5, depending upon the scenario, and the minimum standard deviation allowed is 2. In this approach, as a uni-model distribution is used, it lacks robustness when faced with multi-modal backgrounds where the means of the distributions differ greatly. In addition, although the approach is robust to rapidly changing illumination conditions, the technique is less robust to moved background objects, due to the slow update parameters (but only if the background objects appear within the volume of interest (VOI) defined in section 5.2.1). However, neither of these scenarios occurred within any of the evaluation test dataset scenarios.



(a) (b) (c) (d) (e)

Figure 4.14: Background subtraction; (a) Background image; (b) Background edges; (c) Foreground image; (d) Foreground edges after post-processing; (e) Background disparity model.

Figure 4.14(b) shows the background gradient model that has been initialised from figure 4.14(a) (the blue pixels define background gradients that have a component greater than $min_{grad} = 10$). The extracted foreground gradients from figure 4.14(c) can be seen in figure 4.14(d) (where the red pixels define foreground gradients that have a component greater than $min_{grad} = 10$).

4.3.6.2 Background Disparity Model

The background disparity model is similar to that of the background gradient model. However, the model is not based on a uni-modal distribution, instead foreground disparity pixels are declared if there is *any* difference between the disparity obtained from the current image and background disparity model. In addition, the initial background disparity is set to that of the scene ground-plane. This background model is used exclusively by the BGCPs extraction technique. Figure 4.14(e) illustrates a typical background disparity model created for use for one of the dataset test sequences.

4.3.6.3 BGCPs

Using these two background models, it is possible to extract Background GCPs (BGCPs). The first step in this process removes FARs that are explained by both the background disparity model and the background gradient model, so for example if a pixel, (x, y), is part of a FAR but *not* declared as a foreground gradient value and

$$SSD(rgb_1^{(x,y)}, rgb_2^{(x+d,y)}) <= 3 \times (t_{FARs}^{rgb})^2 \text{ , and} \quad (4.7)$$

$$SSD(grad_1^{(x,y)}, grad_2^{(x+d,y)}) <= 6 \times (t_{FARs}^{grad})^2 \quad (4.8)$$

where d is defined as the background disparity at (x, y). Then the FAR pixel is declared as being explained by the background models, and so is no longer declared as a FAR.

After this process, strong background gradients edges (that have a component greater than $min_{grad} = 10$) are obtained from I_1, using non-maximal suppression of the background gradients (similar to the technique employed in section 4.3.4.1). The image is then searched to find two background edges that are separated vertically by a region of homogeneity that;

1. does *not* contain a FAR;
2. does *not* contain a GCP with a disparity different to the background disparity model;
3. does *not* contain any pixel that fails to pass the inequalities 4.7 and 4.8.

If such a region is found between two background gradients edges, then a BGCP is created at each pixel between the two edges, whereby the disparity of each BGCP is set to that of the background disparity model. A BGCP region can then be propagated towards the groundplane in a similar manner to GCP regions one scanline at a time *iff* each pixel in the BGCP region on the next scanline adheres to the three constraints outlined above. Finally, a propagated BGCP region can extend left and right until a maximum vertical edge is reached or any one of the three constraints fails.

BGCP regions are handled in the same manner as GCP regions in the dynamic programming algorithm. Figure 4.15 shows some improvements to dense disparity maps that are possible using BGCPs. In particular notice how the areas circled in red in figure 4.15(d) are improved in (e).

4.3.7 Dynamic Programming

The technique applied to obtain dense disparity from the groundplane space images is based on a *dynamic programming* (DP) approach. As described in section 4.2.4.3, this technique is a hybrid of global and local based optimisation techniques, whereby a global optimisation of each pair of matching epipolar lines, or scanlines, are treated independently. The correspondence problem between images can then be cast as a problem of finding a minimum solution to a global scanline cost function. Dynamic programming, which is a way of efficiently minimising functions of a large number of discrete variables [176], is then applied to find a solution.

(a) (b) (c) (d) (e)

Figure 4.15: The advantages of using BGCPs; (a) Input image; (b) GCP regions; (c) BGCPs; (d) Dense disparity without BGCPs (areas that can be improved using BGCPs highlighted by a red ellipse); (e) Dense disparity with BGCPs.

4.3.7.1 Pruning the search space

Although the computation cost for DP techniques can be significantly less than global techniques, it can still be relatively expensive if a large disparity limit is applied. When formulated as a DP problem, finding the best path through the search space of width N and disparity range D requires considering $N \times D$ dynamic programming nodes (each node being a potential place along a path) [18]. In some of the evaluation test dataset images, for example within the synthetic dataset, the computation of each scanline must consider $1,024 \times 75 = 76,800$ nodes. A reduction in this search space is advantageous for two reasons; firstly potential disparity mis-matches are removed, which may lead to more robust disparity estimation; and secondly computational efficiency can be improved by reducing the overall number of paths through the search space.

In the proposed approach, this search space can is significantly reduced using three techniques. The first technique, as previously described is to find the disparity of the input images in ground-plane space, therefore eliminating disparities that represent point located below the scene ground-

plane. The second technique employs the idea of a *dynamic* disparity limit constraint. In the proposed approach, the limit for a given scanline is determined as the maximum of the previous scanline's *dense* foreground disparity, the current and next scanlines GCP disparity, and the maximum disparity within the corresponding three scanlines in the background model. The final technique is to remove nodes for which the matching cost is too great, therefore allowing no dynamic programming path to pass through them. A potential match between (x, y) and $(x, y + d)$ can be removed if

$$SSD(rgb_1^{(x,y)}, rgb_2^{(x+d(x,y),y)}) > 3 \times (t_{DP}^{rgb})^2 \text{ , or} \tag{4.9}$$

$$SSD(grad_1^{(x,y)}, grad_2^{(x+d(x,y),y)}) > 6 \times (t_{DP}^{grad})^2 \tag{4.10}$$

where y is the current scanline and $d(x, y)$ is the disparity at pixel (x, y).

4.3.7.2 Scanline Cost Function

The dynamic programming algorithm optimises each scanline by minimising the cost function

$$\begin{aligned} C(x, y, d(x, y)) = \quad & \sum_x M_{cost}(x, y, d(x, y)) + \\ & \sum_x H_{cost}(d(x, y), d(x + 1, y)) + \\ & \sum_x V_{cost}(d(x, y), d(x, y - 1)) \end{aligned}$$

that is composed of three components; a matching cost function, M_{cost}; a horizontal smoothness cost, H_{cost}; and a vertical smoothness cost, V_{cost}. This vertical smoothness cost is employed to enforce inter-scanline consistency, which helps reduce artifacts such as streaking effects in the final disparity map.

In the approach, $M_{cost}(x, y, d(x, y))$ is the cost of matching pixel (x, y) in I_1 to pixel $(x + d(x, y), y)$ in $H_{12}I_2$. This is defined as either $SSD(rgb_1^{(x,y)}, rgb_2^{(x+d(x,y),y)})$ or $SSD(grad_1^{(x,y)}, grad_2^{(x+d(x,y),y)})$, depending on which is the larger value[4]. This cost function therefore automatically switches between a colour or gradient based cost function, depending on which is the most relevant. This can be advantageous at object edges, whereby the difference in colour intensities between pixels can be small, but the gradient difference may be large.

[4]It should be noted that gradients are not directly compared to pixel intensities. Rather the technique resorts to using solely gradients *iff* it reduces ambiguities that were present within the comparison of pixel intensities.

The horizontal and vertical cost functions are defined as

$$H_{cost}(d(x,y),d(x+1,y)) = \begin{cases} \alpha^2 & \text{if } |d(x,y)-d(x+1,y)|>0; \\ 0 & \text{otherwise.} \end{cases}$$

$$V_{cost}(d(x,y),d(x,y-1)) = \begin{cases} 0 & \text{if } d(x,y)=0 \text{ and } (x,y) \text{ is not a GCP;} \\ \beta^2 & \text{else if } |d(x,y)-d(x+1,y)|>0 \text{ and } d(x,y-1)=0; \\ \beta^2 & \text{else if } |d(x,y)-d(x+1,y)|>1 \text{ and } d(x,y-1)>0; \\ 0 & \text{otherwise.} \end{cases}$$

The H_{cost} function adds a constant cost value α^2 to $C(x,y,d(x,y))$ for every disparity discontinuity in the horizontal direction. The V_{cost} function, however, is slightly more sophisticated. When in groundplane space, the disparity of the groundplane at any point in the input images is zero, and any point off the groundplane has a disparity greater than zero. For all objects that are above the groundplane, the disparity naturally varies slightly between scanlines (assuming that the stereo camera rig is positioned upright, or at an overhead viewpoint) as the objects height gradually rises and falls in height. Therefore, the V_{cost} function adds a constant cost value β^2 to $C(x,y,d(x,y))$ if; (a) there is any change in disparity and the previous lines disparity was on the groundplane; or (b) if the previous lines disparity was off the groundplane and there is a disparity discontinuity that is deemed to be more than a slight variation caused by the gradual rises and fall in height (in this case if the disparity discontinuities is greater than 1 pixel).

A final case exists where the V_{cost} function equals 0, independently of all other cases, when $d(x,y)=0$ and (x,y) is not a GCP (or a GCP region, or a BGCP). This case exists as, in general, if there is no guidance by GCPs and the disparity of pixels becomes ambiguous, for example in a homogeneous regions where a number of cost paths through the solution space are at a similar cost, it is preferable to return to the disparity of the groundplane. The main reason for this choice shall become clear when the proposed pedestrian detection module is examined in the following chapter. In that module, if ambiguous pixels are set to the groundplane disparity then they can have no influence, in either computational expense or algorithmically, upon the pedestrian detection technique. Finally, it should be noted that if the pixel $(x-1,y)$ does not exist within the range of I_1, then H_{cost} always takes the value of zero, a similar constraint exists for V_{cost} if $(x,y-1)$ is outside the bounds of I_1.

4.3.7.3 Enforcing GCPs

In this dynamic programming approach, GCPs are used in a similar manner to that of [18] in that the value of the matching cost function, M_{cost}, is altered to increase the likelihood of the DP algorithm choosing a path that runs through the GCPs. In this approach, if a pixel is a GCP, GCP region or BGCP then $M_{cost}(x, y, d^{GCP}(x, y)) = 0$, where $d^{GCP}(x, y)$ is the disparity defined by the relevant control point. To ensure that a control point is more likely to be incorporated within the final dense disparity map, a larger value for M_{cost} is assigned to all paths which does not assign the disparity $d^{GCP}(x, y)$ to the pixel (x, y). However, in the proposed approach, the assignment of this larger value for M_{cost} depends on the *type* of control point disparity. GCPs are treated slightly differently than GCP regions and BGCPs, as GCPs can be viewed to be highly reliable matches and as such their disparities should be more stringently enforced, whereas the other two are intended solely as guidance for the disparity within ambiguous regions.

As such, for a GCP pixel the cost for all other disparities is set to a high enough value to ensure that a path through the GCP is chosen. This means that the absolute minimum cost for all other disparities should be $1 + \beta^2 + 2\alpha^2$, which is the cost of two horizontal disparity jumps plus one vertical disparity jump plus 1 (as to jump to and from a GCP may invoke a total smoothness cost of $\beta^2 + 2\alpha^2$). However, it is noted that this minimum cost should be higher than this absolute minimum, as a number of smaller disparity jumps may be taken when descending from the GCP disparity. In theory, the cost value should be set to ∞, however due to the possible overflow of computer language primitives this value is set to $(t_{DP}^{rgb})^2$ in the implementation of this approach.

It should be noted, however, unlike the technique of [18] the DP path is *not* forced through any GCP disparity values. Using the proposed technique, priority is given to the path through the search space matching as many pixels as possible, rather than passing through as many GCPs as possible. Therefore, for example, if a GCP recommends a large jump in disparity, but this jump (and the ordering constraint) means that a pixel fails to pass the inequalities 4.9 and 4.10 for all possible disparities, then a different path is chosen which matches a higher number of pixels. This design feature is incorporated to ensure that if any false-positive control point disparities exist, the algorithm can recover and ignore the GCP if necessary.

Finally, when dealing with GCP regions and BGCPs the cost for all non control point disparities are set to values significantly less than those for GCPs. In the proposed approach, they are set to be the maximum between the original cost (obtained using the SSD approach), or that of

β^2, which is the vertical smoothness cost. This minimum cost value is chosen as it has little effect unless; (1) there are a number of control points grouped together (in our experiments as $\alpha = 20$ and $\beta = 10$ and, as two jumps in disparity may be required, $2\alpha^2 = 8\beta^2$ therefore 8 or more control points in a row may be needed to create a change in the resultant disparity map, this however may be more or less than 8 depending on the previous lines disparities and the number of disparity discontinuities needed to pass through the control point section); and (2) there is little texture in the region, as the value of β^2 in the proposed approach can be significantly less than the value of a mismatched M_{cost}. Therefore the DP algorithm matches a pixel based on colour and gradients in textured information, but tends towards the disparity of these control point regions if a number of them exist in homogeneous areas.

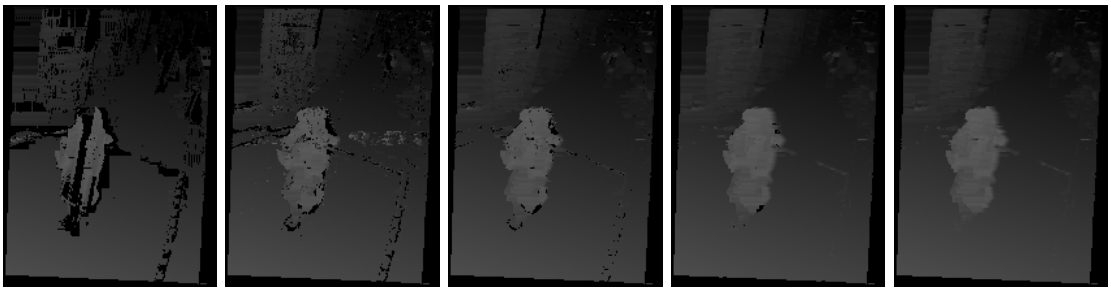
4.3.7.4 Post-processing

It may not be possible to match each point from the two input images due to occlusion issues. A final post-processing step is thus employed to post-process the final disparity map to assign a disparity to each unmatched pixel. This technique simply traverses each scanline of the resultant disparity map for each missing disparity, obtains the closest existing disparity to the left and right of that point, and assigns the disparity to the minimum of those two disparities. This technique, however, is only implemented in this chapter for analysis reasons and is generally not required for input disparity maps in the pedestrian detection module.

4.3.7.5 Parameter Selection

The values of the user defined thresholds, t_{DP}^{rgb} and t_{DP}^{grad}, within the DP algorithm are relatively insensitive once they are set to value high enough values to ensure that correct matches between I_1 and $H_{12}I_2$ are allowed, see figure 4.16 (note that these disparity maps have not been post-processed). For the disparity matches in figure 4.16(b) the thresholds are set lower than the respective GCP thresholds, and as such some of the GCPs are rejected as mis-matches. In figure 4.16(c) the thresholds are increased to that of the GCP thresholds, and as such a number of incorrect matches are removed. The disparity estimation result presented in figure 4.16(d) illustrates the results with the threshold used in our experiments, these values ensure that the vast majority of pixels are correctly matched, while rejecting incorrect matches.

The results obtained from a variety of smoothness costs, α and β, can be seen in figure 4.17. In the first row of this figure, the vertical cost is removed and a variety of values for α are set. When



(a) (b) (c) (d) (e)

Figure 4.16: DP cost parameters (Note: $t^{rgb}_{FARs} = 20$, $t^{grad}_{FARs} = 10$, $t^{rgb}_{GCPs} = 20$, $t^{grad}_{GCPs} = 10$); (a) Input control points; (b) $t^{rgb}_{DP} = 10$, $t^{grad}_{DP} = 5$; (c) $t^{rgb}_{DP} = 20$, $t^{grad}_{DP} = 10$; (d) $t^{rgb}_{DP} = 50$, $t^{grad}_{DP} = 25$; (e) $t^{rgb}_{DP} = \infty$, $t^{grad}_{DP} = \infty$.

α is also set to zero the disparity can become noisy, especially at regions where there are no control points to guide results. As the value of α is increased, so is the amount of horizontal smoothing. In row one of figure 4.17, α is increased up to a point where the resultant disparity map becomes over smoothed. The second row of figure 4.17 shows the effects of setting α to zero and varying the horizontal smoothing cost. It can be seen that over-smoothing in the vertical direction can occur quickly – even at $\beta = 12.5$ some over smoothing has occurred at the top of the pedestrians head. Over smoothing is both directions can be reduced by incorporating both smoothness functions into the algorithm, see row three of figure 4.17, where the reduction in both types of over smoothing can be seen, for example when $\alpha = \beta = 250$. However, as vertical over-smoothing tends to occur at lower values than that of horizontal over-smoothing, by making $\beta < \alpha$ the overall algorithm can be made more robust, see row 4 in figure 4.17. In our experiments, $\alpha = 20$ and $\beta = 10$, however it has been found that when using $\beta = 0.5\alpha$ the choice of values are not highly critical, for example using the disparity map obtained using $\alpha = 250$ and $\beta = 125$ still results in the correct detection of the person using the proposed pedestrian detection module. An overview of all parameters using in the proposed disparity estimation technique is presented in table 4.1.

4.4 Experimental Results

In this section, results of the proposed technique on a synthetic dataset are quantitatively and objectively evaluated and the approach is shown to outperform a variety of other disparity estimation algorithms for the given test set. In addition, visual evaluation against a variety of indoor and outdoor scenarios involving a number of camera orientations, background scenes and lighting conditions is carried out. This subjective visual evaluation reinforces the results obtained using

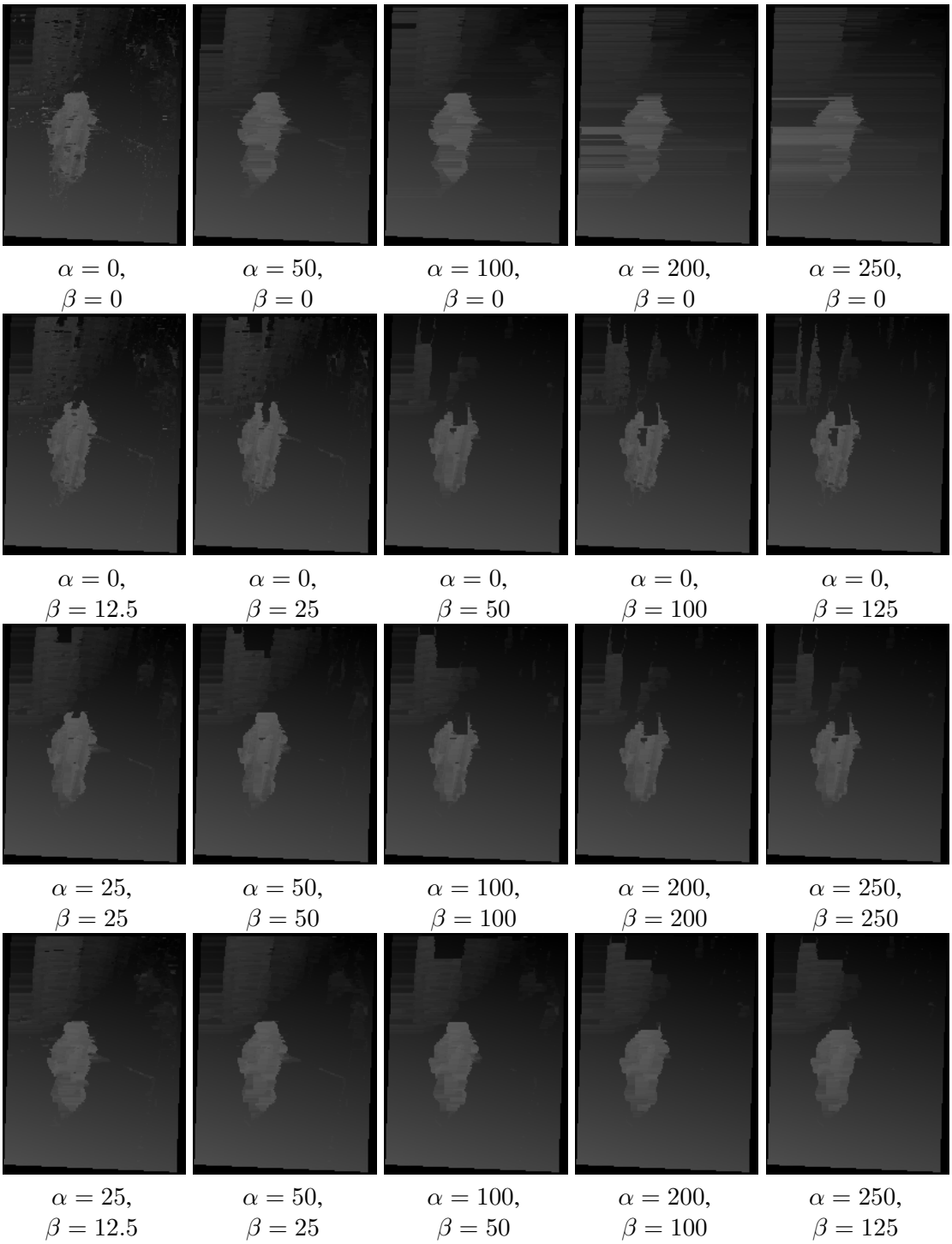


Figure 4.17: DP smoothness values (Note: $t^{rgb}_{FARs} = 20$, $t^{grad}_{FARs} = 10$, $t^{rgb}_{GCPs} = 20$, $t^{grad}_{GCPs} = 10$, $t^{rgb}_{DP} = 50$, $t^{grad}_{DP} = 25$); (Row 1) Altering α; (Row 2) Altering β; (Row 3) Setting $\alpha = \beta$; (Row 4) Setting $\alpha = 0.5\beta$.

Parameter	Value	Overview
t_{FARs}^{rgb}	20 10	– used to determine FARs from which GCPs are extracted
t_{FARs}^{grad}	10 10	– most sensitive thresholds – if too high the number of GCPs extracted is constrained – if too low the propagation of BGCPs is constrained
t_{GCPs}^{rgb}	40 20	– used to determine possible GCPs matches
t_{GCPs}^{grad}	20 10	– relatively insensitive once high enough to allow correct matches
t_{DP}^{rgb}	50 30	– used to determine possible DP matches
t_{DP}^{grad}	25 15	– relatively insensitive once high enough to allow correct matches
α	20 10	– horizontal smoothing cost – relatively insensitive when $\beta = 0.5\alpha$
β	10 5	– vertical smoothing cost – relatively insensitive once lower than α, usually set to 0.5α
min_{grad}	10 10	– used to define if a gradient is large enough to be considered as a GCP match
n	500 500	– the number of frames that a foreground pixel must remain in the cache before a background model is updated
m	4/5 4	– the standard deviation for the background gradient model. The darker the image sequence, the less the value of m. For the real-world datasets; $m = 4$ for the *Corridor* and *Grafton* 3 sequences; and $m = 5$ for all the other sequences.

Table 4.1: Parameter Selection. Two figures, A B, are provided in the Value column. A represents the parameter values for the real-world datasets, B represents the parameter values for the synthetic dataset. This difference in parameter values is due to the colour values in the synthetic dataset only spanning $\approx 70\%$ of the colour spectrum range.

the synthetic dataset.

4.4.1 Digiclops Stereo Camera Rig

In all the experiments undertaken in this work, the camera rig used to obtain the input data is the Digiclops stereo camera system [169] manufactured by Point Grey Research [170]. This stereo rig consists of three digital cameras, each with a focal length of 3.8mm, aligned at the edge points of a right angled triangle, see figure 4.18. In our experiments only two of the three cameras are employed, where the baseline between the cameras is 100mm. For the majority of the evaluation test scenarios, the input camera images are rectified through the Triclops stereo vision library that is packaged with the Digiclops stereo vision camera system. This rectification technique also removes radial distortion and the resultant images fit an ideal stereo camera model to within 0.06 pixels [170]. This rectification technique, however, was not possible for the outdoor scenes whereby the images are rectified using the technique outlined in [154].

Figure 4.18: Digiclops stereo camera rig.

4.4.2 Synthetic Dataset Generation

In order to quantitatively evaluate the proposed disparity estimation approach, a dataset with groundtruth disparities is required. Unfortunately standard disparity datasets, such as the *Tsukuba*, *Venus*, or *Map* datasets [204, 171] (see figure 4.19) are not applicable for evaluation of the proposed disparity estimation approach, mainly due to the lack of a groundplane region within the scene and the inability to obtain the background models required to obtain BGCPs. In addition, as the proposed disparity estimation technique is designed for pedestrianised scenes, it would be advantageous to determine the robustness of the approach to a variety of challenging scenarios such as different lighting conditions, shadows, a lack of texture at depth discontinuities, homogeneous foreground and background regions, and most importantly pedestrians exhibiting a variety clothing, orientations, distances and scales.

For these reasons a new synthetic dataset, designed to incorporate a number of difficulties associated with typical pedestrianised scenarios, was created using computer graphics techniques in 3D Studio Max. Each 3D scene was designed to incorporate a flat groundplane, one or more background objects, and a number of varying pedestrian models, see figure 4.20. Throughout the dataset, a variety of texture maps were chosen for the groundplane. These textures vary from textured or tiled surfaces through to a single homogeneous colour. Finally, a variety of ambient and directional lighting sources were introduced, designed to mimic the lighting conditions of both indoor and outdoor scenarios. Depending on the lighting conditions, shadows (both cast- and self-shadows) range from subtle to strong.

For each 3D scene, a virtual stereo camera rig, designed to emulate the Digiclops camera setup was created. One minor alteration was to increase the image size from 640×480 to 1024×614 pixels to incorporate more objects into a given image rendering. Each 3D scene was then rendered from each virtual camera viewpoint twice; the first rendering contained no foreground objects,

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

Figure 4.19: Standard synthetic groundtruth disparity datasets; (a) *Tsukuba*; (b) *Tsukuba* disparity; (c) *Venus*; (d) *Venus* disparity; (e) *Map*; (f) *Map* disparity.

(a) (b) (c) (d)

Figure 4.20: Synthetic data generation; (a) 3D model wireframes; (b) Rendered scene; (c) Groundtruth depth map; (d) Groundtruth disparity map.

and as such can be used to initialise the background models; the second incorporated a number of foreground pedestrians. Each rendering of the scene was accompanied by a groundtruth depth map, see figure 4.20(c), which is obtained by rendering the scene's depth *z-buffer*. From this depth map the groundtruth disparity of each pixel can be calculated via

$$d = \frac{b * f_p}{z}$$

where d and f_p are the disparity and focal length in pixels respectively, b is the stereo camera baseline in Euclidean co-ordinates, and z is the Euclidean depth of the 3D point from the camera. The value of f_p is calculated via

$$f_p = \frac{f_e * w}{a}$$

where f_e is the Euclidean focal length, w is the image width in pixels, and a is the Euclidean width of the aperture (or the Euclidean width of the Charge Coupled Device (CCD) sensor in digital cameras). Each of the 8 synthetic scenes (as rendered from camera C_1) can be seen in figure 4.21, where rows 1 to 8 correspond to synthetic scenes 1 to 8 respectively. In this figure Bn represents the rendering of scene n with only background objects, and Fn represents the rendering of scene n with foreground objects.

(a) (b) (c) (d)

Figure 4.21: Synthetic scenes 1-8; (a) Background scenes, (Row 1) $B1$ to (Row 8) $B8$; (b) $B1$ to $B8$ groundtruth disparities; (c) Foreground scenes, (Row 1) $F1$ to (Row 8) $F8$; (d) $F1$ to $F8$ groundtruth disparities.

4.4.3 Disparity Estimation Evaluation

4.4.3.1 Quantitative Evaluation

Using the 16 synthetic scenes ($B1$–$B8$ and $F1$–$F8$) a quantitative comparison of the proposed technique to other disparity estimation techniques was undertaken using the Middlebury College open source stereo algorithm evaluation test bed [205]. This framework provides a stand-alone C++ implementation of many stereo algorithms, from which a large variety of algorithms were applied to each of the 16 scenes. In addition, it provides a module for the quantitative evaluation of disparity results. From this evaluation module, two standard metrics (described below) were chosen to evaluate the proposed technique against the implementation of 700 disparity estimation algorithms provided by the Middlebury framework.

During the evaluation process global-based optimisation techniques, such as graph-cuts, belief propagation and simulated annealing, were eliminated due to the huge computational complexity involved in obtaining a result from the 1024×614 images, resulting in processing times of between 20 minutes to several hours for a single disparity map. For the experimental evaluation, all the test algorithms were provided with a maximum disparity of 75 pixels (the maximum disparity in the test set being 72 pixels), except for the proposed technique, which applied the dynamic based disparity limit approach with an *absolute* maximum disparity of 150 pixels (used solely for memory allocation purposes within the software implementation of the framework). Using the Middlebury framework three types of optimisation were tested; local based techniques based on Winner-Takes-All (WTA) optimisation; Dynamic Programming (DP); and Scanline Optimisation (SO), which obtains the minimum path through a cost function obtained between two corresponding scanlines. It should be noted that the DP algorithm applied in the Middlebury evaluation differs from the implementation in this work, as with the proposed approach a horizontal smoothness cost is included, but not vertical smoothness cost. In addition, they incorporate an occlusion cost, which is a constant value charged to each pixel that has no associated match.

For each optimisation technique both SSD and SAD cost functions were evaluated. For the local based techniques a large variety of aggregation sizes and techniques, such as square windows, shiftable windows [18], binomial filters, regular diffusion, and membrane diffusion [193], were tested. For the DP and SO techniques a variety of occlusion and smoothness costs were tested, ranging from 20 to 700. In addition, a variety of cost truncation values were incorporated into the algorithms, ranging from 1 to ∞. This truncation value can limit the influence of incorrect

matches near depth discontinuities [171]. In total 700 disparity estimation algorithm variations for each of the 16 synthetic scenes were evaluated using the Middlebury framework; 350 of these were local based techniques; 200 were DP approaches; and 150 were SO algorithms.

In addition, the proposed approach was evaluated using the Middlebury evaluation module for each of the 16 synthetic scenes. For the implementation of the proposed technique, the groundplane homography was obtained by manually selecting 3 corresponding groundplane feature points. The parameters set were then set to $t_{FARs}^{rgb} = 10$, $t_{FARs}^{grad} = 10$, $t_{GCPs}^{rgb} = 20$, $t_{GCPs}^{grad} = 10$, $t_{DP}^{rgb} = 30$, $t_{DP}^{grad} = 15$, $\alpha = 10$, $\beta = 5$. It should be noted that these threshold values were set slightly lower than the corresponding thresholds used when calculating the disparity from real-world data. This reduction was implemented as the colour values in the synthetic dataset only spans $\approx 70\%$ of the colour spectrum range. Therefore, as the maximum absolute difference between any two given pixels is reduced, allowing the both the threshold and smoothness costs to be slightly lowered. A visual comparison of the resultant disparity maps against the best performing local and DP based approaches as outlined below can be viewed in figures 4.24 and 4.25.

RMS Error For quantitative evaluation of the results two standard quality metrics, taken from [171], were employed to gauge the performance of the stereo correspondence algorithms. The first, and most important, metric used was the RMS (root-mean-squared) disparity error between a computed disparity map $d_c(x, y)$ and the groundtruth map $d_t(x, y)$. This is obtained via [171]

$$\text{RMS error} = \left(\frac{1}{N} \sum_{(x,y)} |d_c(x, y) - d_t(x, y)|^2 \right)^{\frac{1}{2}} \tag{4.11}$$

where N is the total number of pixels. To evaluate the performance of a particular algorithm, the average RMS error over the 16 synthetic data was calculated. These average RMS error values were then used to obtain a ranking for each one of the 700 Middlebury algorithms. The top ranking local based technique was *SSD_sw35_t0010* with an average RMS error of 5.73, meaning that the algorithm applied a SSD cost function that was aggregated over a 35 pixel sized shiftable window [18] with a truncation value of 10.

The top ranking non-local based technique was *SSD_DP_o060_s040_t0010* with an average RMS error of 3.44. This was a dynamic programming algorithm using a SSD cost function, an occlusion cost of 60, a smoothness cost of 40, and a truncation value of 10. The RMS error of

Technique	*Average RMS error*
SSD_sw35_t0010	5.73
SSD_DP_o060_s040_t0010	3.44
Minimum RMS error result	3.17
Proposed Technique	1.55

(a)

Technique	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*	*B8*
SSD_sw35_t0010	0.91	0.90	16.75	16.51	1.27	1.31	0.98	6.20
SSD_DP_o060_s040_t0010	0.76	0.95	8.48	8.65	0.71	1.08	0.53	1.19
Minimum RMS error result	0.51	0.58	8.48	7.65	0.50	0.49	0.48	1.18
Proposed Technique	0.52	0.95	0.72	0.71	0.61	0.57	0.52	0.79

(b)

Technique	*F1*	*F2*	*F3*	*F4*	*F5*	*F6*	*F7*	*F8*
SSD_sw35_t0010	3.53	3.92	13.91	8.95	2.84	4.21	3.21	6.29
SSD_DP_o060_s040_t0010	3.35	3.60	6.64	7.09	3.02	3.47	3.01	2.50
Minimum RMS error result	3.01	3.55	6.32	7.09	2.56	3.15	2.65	2.46
Proposed Technique	2.54	3.35	2.06	2.85	1.99	2.42	2.32	1.81

(c)

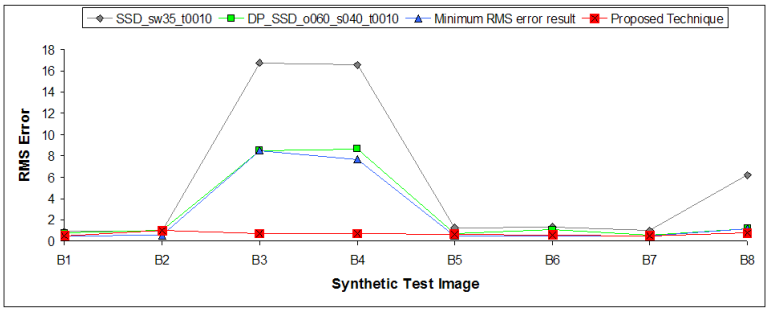
Table 4.2: Synthetic data RMS error results; (a) Average RMS error for all 16 synthetic scenes; (b) RMS error results for *B1* to *B8*; (c) RMS error results for *F1* to *F8*.

these two algorithms for each of the 16 synthetic scenes can be viewed in table 4.2. This table also includes the minimum RMS error result obtained using all 700 algorithms in each 16 scenes, and the results of the proposed algorithm on the same dataset. These RMS error results are also graphed in figure 4.22.

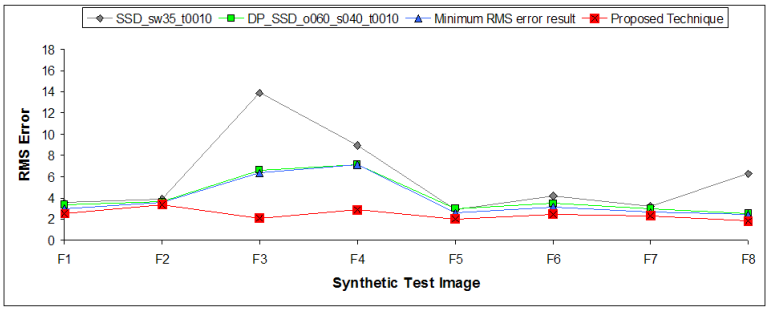
Bad Pixel % The RMS error metric alone is not sufficient to give a full indication of algorithmic performance as it can be contaminated by a small number of erroneous pixels with large disparity errors. Therefore a second metric, the percentage of badly matching pixels, was calculated for each of the top performing algorithms. For this metric, a pixel is defined as being badly matched if the difference between $d_c(x, y)$ and $d_t(x, y)$ is outside some disparity error tolerance, δ [171] . Therefore

$$\text{Bad Pixel } \% = \frac{1}{N} \sum_{(x,y)} (|d_c(x, y) - d_t(x, y)| > \delta) \tag{4.12}$$

An issue with this metric, however, is the choice of δ. If it is set too high, then all disparity techniques may return a % value of near 0, resulting in a lack of distinguishing information between the algorithms. A similar effect occurs if % is too low, as the resultant metrics from all techniques will tend towards 100. Ideally, the value setting of δ should reflect the inherent difficulty associ-



(a)



(b)

Figure 4.22: Synthetic data RMS error results; (a) RMS error results for *B1* to *B8*; (b) RMS error results for *F1* to *F8*.

ated with the input dataset. For clean, noise-free images that are unrealistically easy to solve, a low value of δ should be set. For more complicated datasets, more information can be obtained from the metric if the value of δ is set to a higher level. In our experiments, δ is set to 3.17, which is the average RMS error of the *minimum* results obtained for each of the 16 synthetic images using the 700 Middlebury algorithms, see table 4.22(a). This value of δ is selected as it can be seen as representative of the inherent difficulty associated with the synthetic dataset. As the RMS error is mathematically the spatial equivalent to standard deviation, using this value of δ the Bad Pixel % can be viewed as the percentage of pixels outside one standard deviation of the minimum average best RMS error results for the 16 synthetic images. The results of the Bad Pixel % metric for the best performing RMS algorithms can be viewed in table 4.3 and graphed in figure 4.23. Note that in figure 4.23(a), although the maximum % of bad pixels in the results are 77.38%, the graph focuses on the values less than or equal to 10% to increase the level of detail within the more informative areas of the graph.

Evaluation Using the results of the two metrics a quantitative evaluation of the proposed algorithm can be made. The most obvious improvement in results can be seen in the results from the synthetic scenes 3 and 4, which prove the most difficult for both local and DP based tech-

Technique	*Average Bad Pixel %*
SSD_sw35_t0010	17.76
SSD_DP_o060_s040_t0010	17.82
Minimum RMS error result	16.47
Proposed Technique	3.44

(a)

Technique	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*	*B8*
SSD_sw35_t0010	0.79	0.74	77.38	49.87	0.53	0.55	0.55	4.87
SSD_DP_o060_s040_t0010	0.61	1.22	64.27	54.7	0.57	1.38	0.27	1.76
Minimum RMS error result	0.21	0.44	64.27	56.43	0.24	0.19	0.16	1.68
Proposed Technique	0.10	1.92	1.39	0.31	0.09	0.18	0.51	0.11

(b)

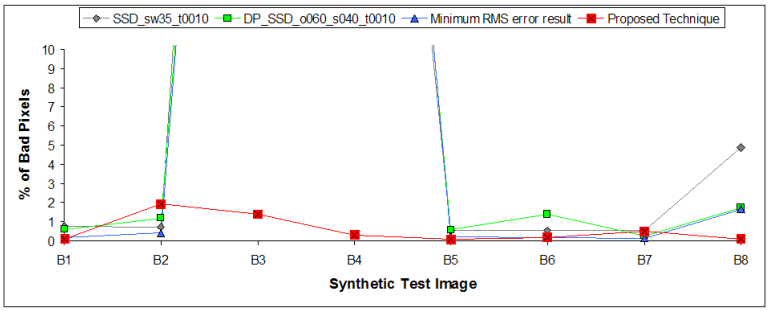
Technique	*F1*	*F2*	*F3*	*F4*	*F5*	*F6*	*F7*	*F8*
SSD_sw35_t0010	11.78	8.82	51.84	28.06	8.06	14.54	10.13	15.63
SSD_DP_o060_s040_t0010	13.91	9.06	51.67	42.59	12.48	10.86	9.88	9.85
Minimum RMS error result	7.65	6.98	49.17	42.59	6.29	10.68	6.77	9.69
Proposed Technique	10.34	7.76	4.89	7.38	4.34	5.33	6.29	4.16

(c)

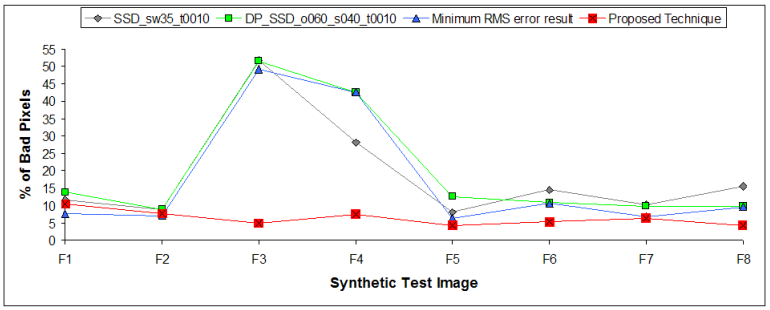
Table 4.3: Synthetic data Bad Pixel % results; (a) Average Bad Pixel % for all 16 synthetic scenes; (b) Bad Pixel % results for *B1* to *B8*; (c) Bad Pixel % results for *F1* to *F8*.

niques, mainly due to the lack of texture on the homogeneously coloured groundplane. For *SSD_sw35_t0010* the window size is not large enough to incorporate sufficient texture information to obtain the correct disparity in the WTA framework, however increasing the window size further can result in a more blocky disparity map, whereupon important details can be lost. For *SSD_DP_o060_s040_t0010* streaking occurs as the smoothness costs are too high, which results in the disparity remaining at some constant (and generally incorrect) value throughout the homogeneous background, however greatly decreasing these smoothness costs can result in a much noisier disparity map, whereby the disparities change more freely. Due to the homogeneous colour the disparity is not guaranteed to be jump to the correct disparity with a lower smoothness cost. Using the proposed technique however gives a reduction in RMS error of 7.76, 6.94, 4.26 and 4.24 from the minimum RMS error results obtained using the Middlebury algorithms for $B3$, $B4$, $F3$ and $F4$ respectively. In addition, a reduction in the Bad Pixel % of 62.88, 56.12, 44.28 and 35.21 for the same scenes can also be obtained.

These results are the most significant in the dataset, however, even without these two scenes the average results from the remaining 12 scenes for the proposed technique outperforms the corresponding value for the minimum results obtained from the Middlebury framework, recording a drop of 0.23 and 0.82 for RMS error and Bad Pixel % respectively. A further breakdown of these figures into foreground and background scenes reveals an equal RMS and a minor rise of 0.04 in



(a)



(b)

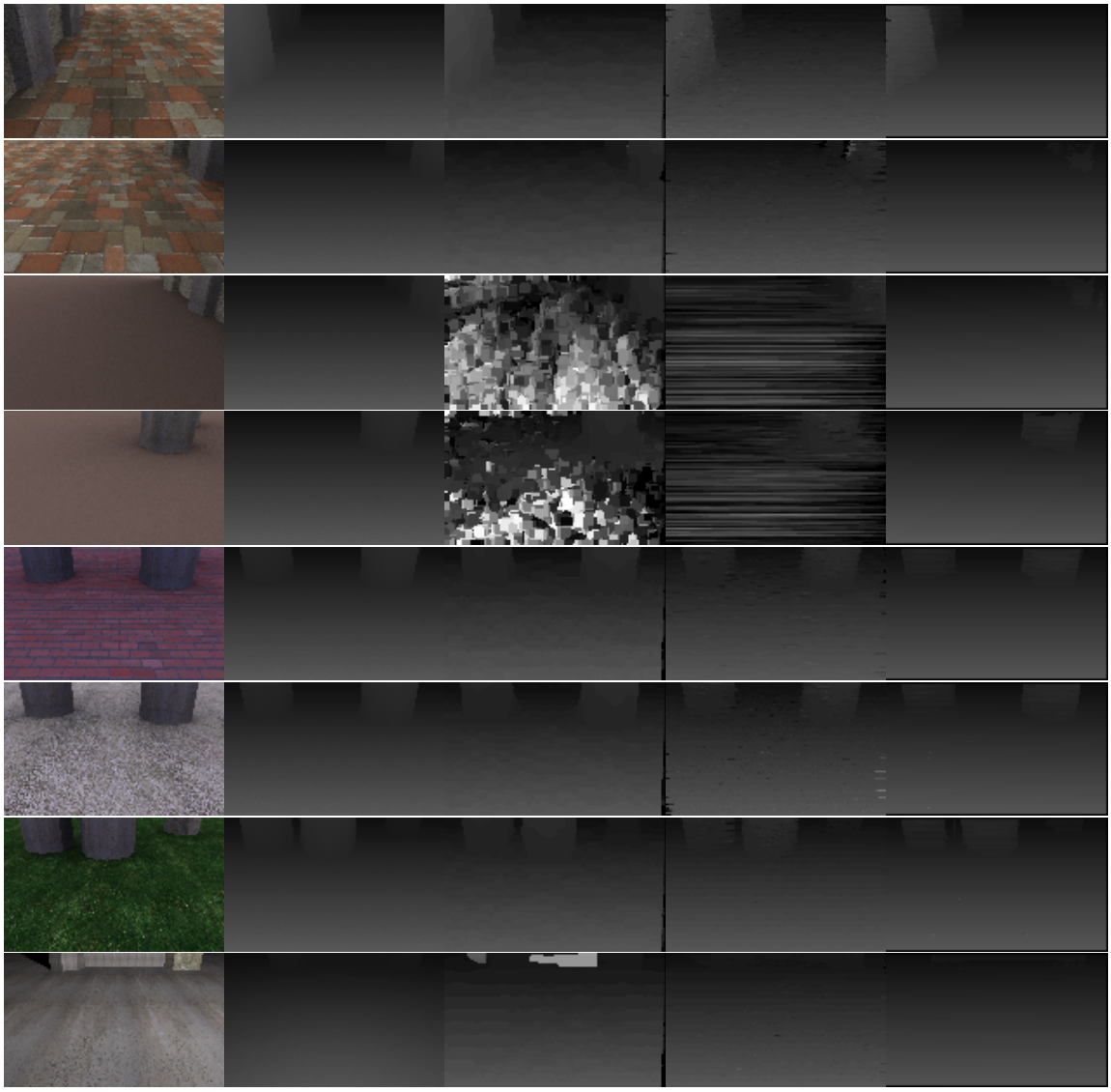
Figure 4.23: Synthetic data Bad Pixel % results; (a) Bad Pixel % results for *B1* to *B8*; (b) Bad Pixel % results for *F1* to *F8*.

Bad Pixel % for the background scenes, but a drop of 0.49 and 1.64 for RMS error and Bad Pixel % respectively for the foreground scenes. These values rise to 1.62 and 13.02 for RMS error and Bad Pixel % if all 8 scenes are incorporated, see tables 4.2(a) and 4.3(a), revealing drops of 1.81 and 14.88 for the background scenes, and drops of 1.43 and 13.02 for the foreground scenes.

For the single best performing algorithm tested, *SSD_DP_o060_s040_t0010*, the proposed approach obtained improved results in *both* RMS error and Bad Pixel % in 16 out of the 18 scenes. The two scenes where it does not make an improvement in both metrics, i.e. $B2$ and $B7$, it increases Bad Pixel % by 0.7 and 0.24 respectively. However, for these scenes the RMS error obtained is almost exactly equal to that of *SSD_DP_o060_s040_t0010*, revealing an equal Bad Pixel % for $B2$ and a drop of 0.01 for $B7$. For the best performing local technique, *SSD_sw35_t0010*, the proposed algorithm obtains improved results in *both* RMS error and Bad Pixel % in all but one scene, $B2$, where an increase in the Bad Pixel % of 1.18 is recorded, again though the RMS error obtained is almost the same as that of *SSD_sw35_t0010*.

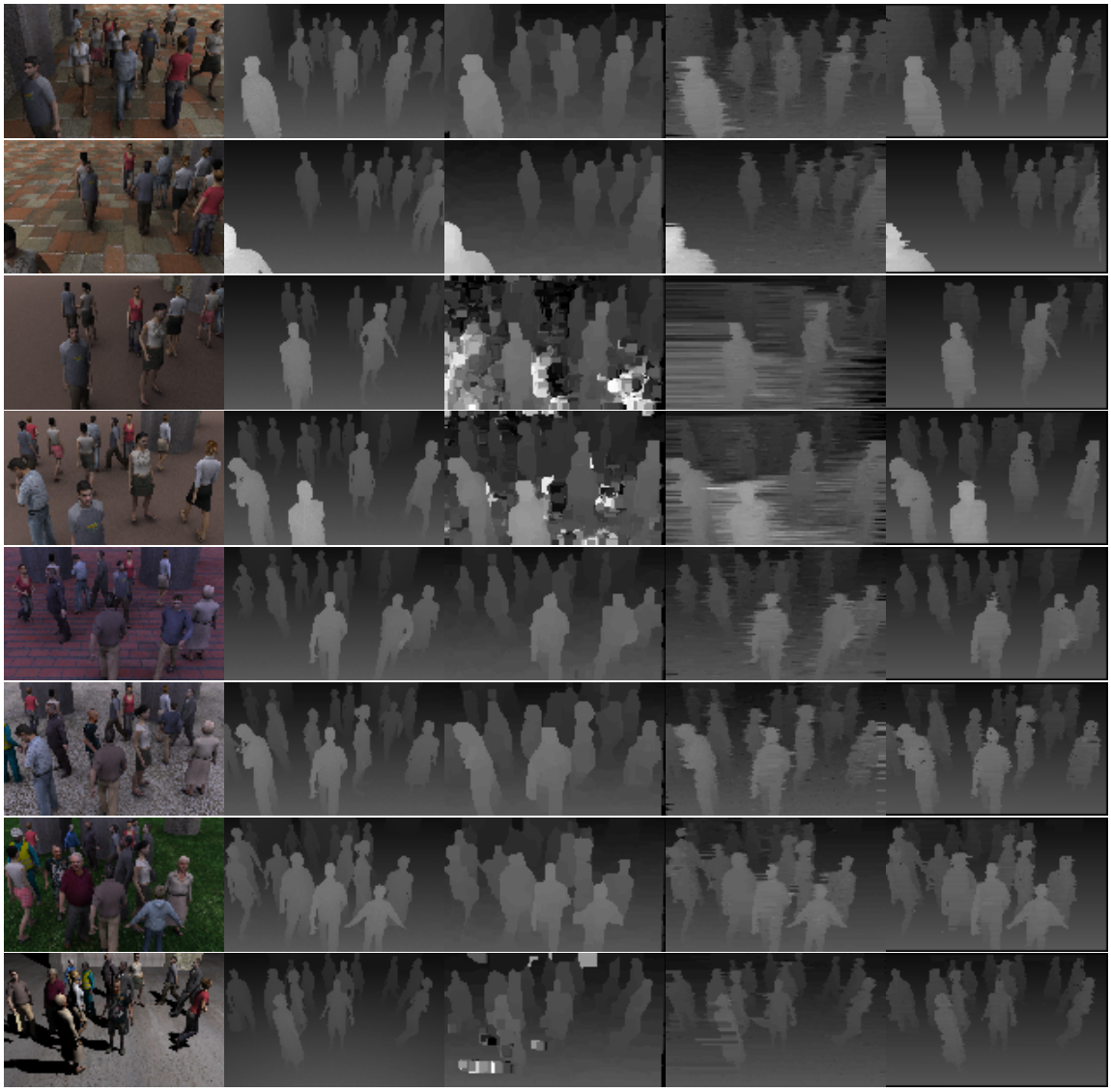
4.4.3.2 Visual Evaluation

From the results in the previous section, the advantages of applying the proposed technique for disparity estimation in pedestrianised scenarios can be seen, especially in the presence of textureless



(a) (b) (c) (d) (e)

Figure 4.24: Synthetic background scene results; (a) Input; (b) Groundtruth disparity; (c) *SSD_sw35_t0010*; (d) *SSD_DP_o060_s040_t0010*; (e) Proposed approach.



(a) (b) (c) (d) (e)

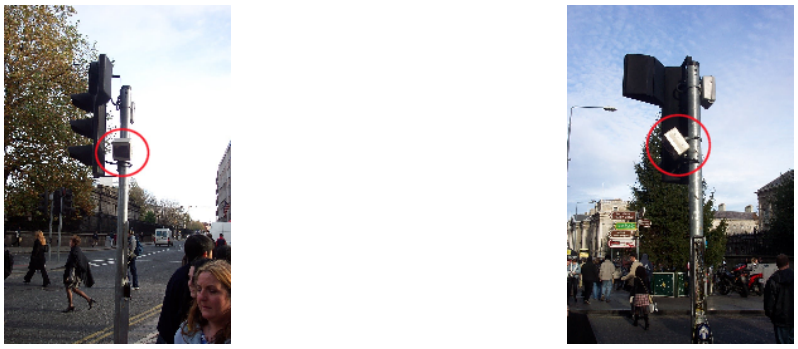
Figure 4.25: Synthetic foreground scene results; (a) Input; (b) Groundtruth disparity; (c) *SSD_sw35_t0010*; (d) *SSD_DP_o060_s040_t0010*; (e) Proposed approach.

background regions. For a number of scenarios the approach is shown to perform robustly and either as well as, or better than that of other disparity estimation algorithms of similar computational expense. However, the evaluation of a disparity estimation technique should not be based solely on synthetic data. The disadvantages associated with evaluation on synthetic data arise from the fact that real cameras are often afflicted with a number of imperfections that are seldom modeled by synthetic data. These imperfections include aliasing, slight misalignment, noise, lens aberrations, and fluctuations in gain and bias [171]. Although a number of steps were taken to ensure that the synthetic data was as photo-realistic as possible while providing an accurate groundtruth, these imperfections were not incorporated into the virtual camera models. For these reasons a visual evaluation of disparities from real-world scenes was undertaken.

Throughout this work 4 test scenarios are used for evaluation that collectively encapsulate varying camera height, camera orientation and environmental conditions. A varying number of test and development sequences were captured from each scenario at a resolution of 640×480 pixels from the Digiclops camera system, captured between 2–5.5Hz. The experimental sequences were chosen to test the proposed technique extensively in several areas, such as disparity estimation, foreground segmentation, pedestrian detection and tracking. None of the test sequences were used in development of the proposed algorithms. In the evaluation section of this chapter, and indeed for the evaluation of the pedestrian detection and tracking modules, sample results from the various scenarios are presented to illustrate both the success and possible failings of the proposed approach. In each sequence, no restrictions or instructions were provided as to where people could go, what they could do or what they could wear.

Indoor Data Sequence Acquisition The first scenario, which is referred to as the *Overhead* scenario, see figure 4.27, was set in an indoor setting with the camera positioned at around 3 metres above the ground. The camera was then orientated back towards the groundplane. The camera rig in this point of view has a limited field of view and due to its proximity with the groundplane it does not encounter significant occlusion problems. The lighting conditions in the scene are stable, brightly illuminated with a highly reflective ground surface.

The second scenario, which is referred to as the *Corridor* scenario, see figure 4.28, was set in an indoor setting with the camera positioned just above 2 metres from the ground. The camera is orientated at 30 degrees towards the groundplane. The lighting conditions are relatively stable, however the ambient lighting of the scene does fluctuate more than the previous scenario due



(a) (b)

Figure 4.26: Mounted Digiclops within a protective casing; (a) Front viewpoint; (b) Side viewpoint.

a number of skylights present within the scene. In addition, the scene's illumination is more challenging than that of the *Overhead* sequence as it is brightly illuminated on one side, and dark on the other side, due to the skylights in the corridor. This can cause a lack of texture in those areas. Finally, the scene contains a staircase on the right hand side, where people can descend and ascend at will.

The third scenario, which is referred to as the *Vicon* scenario (the reason for this is explained in the following chapter), see figure 4.29, was set in an indoor setting with the camera positioned just above 2 metres from the ground. This setup is similar to that of the *Corridor* scene with regards camera placement and orientation. The lighting conditions are stable, however, as with the *Overhead* scene, the groundplane is highly specular and brightly illuminated. In this scenario, a number of people (between 1 and 8) are constrained to a circle of 3.15 and 5.5 metres in width and length respectively.

Outdoor Data Sequence Acquisition The final scenario, which is referred to as the *Grafton* scenario, see figures 4.30 and 4.31, was taken from a camera which, with the help of Dublin County Council, was mounted withing a protective casing at 2.5 metres above the groundplane at a 45 degree angle on a traffic light pole on Grafton Street, a busy pedestrianised shopping street in Dublin city centre, see figure 4.26. The sequences taken from this scenario are of real-world data containing pedestrians from the general public walking during their daily routine. Although a number of sequences were captured from this scenario, with a variety of pedestrian flow and lighting conditions, the sequences for evaluation were deliberately chosen to contain challenging segments, for example groups of people walking in multiple directions or standing still and rapidly changing lighting conditions.

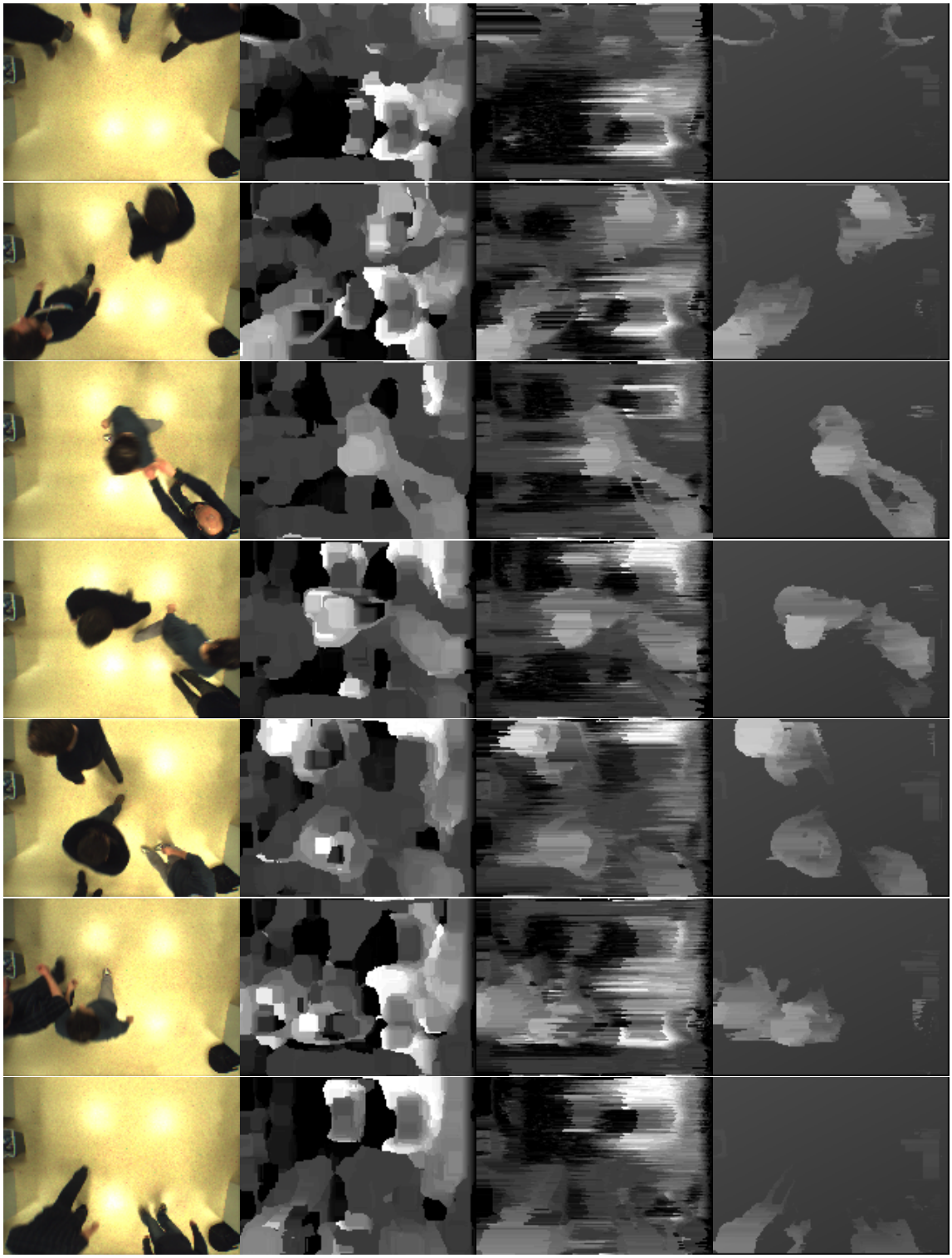
Evaluation For the visual evaluation of the real-world dataset the Middlebury framework was again employed for the comparison against the proposed technique, however, due to the results obtained from the synthetic data results, only DP and local based techniques applying shiftable windows were evaluated. It should also be noted that the input data for the Middlebury algorithms have been both rectified and normalised in colour. From the Middlebury disparity results, one local based technique, namely *SSD_sw35_t1000*, and one DP technique, *SSD_DP_o400_s400_t1000*, were selected that were deemed to have the best overall results from a test dataset consisting of 40 sample images from the four scenarios, where $t1000$ represents an infinite truncation value. It is acknowledged that this selection process is highly subjective, however the results presented are indicative of the best results obtained by *any* one local or DP algorithm from the Middlebury framework. For the implementation of the proposed technique the groundplane homography was, as before, obtained by manually selecting 3 corresponding groundplane feature points. The parameters were set to $t^{rgb}_{FARs} = 20$, $t^{grad}_{FARs} = 10$, $t^{rgb}_{GCPs} = 20$, $t^{grad}_{GCPs} = 10$, $t^{rgb}_{DP} = 50$, $t^{grad}_{DP} = 25$, $\alpha = 20$, $\beta = 10$.

From the resultant disparity maps, the robustness of the proposed technique with regard to homogeneous and highly specular background regions can be seen, especially in the indoor scenes. For example, in the *Vicon* scenario of figure 4.29 row 6, both the local and DP based techniques obtain a substantial erroneous region of disparity due to the lack of texture. This consistently occurs in the both the *Corridor* and most notably in the *Overhead* scenes, whereas the proposed approach treats these homogeneous regions in a consistent manner. It should be noted that in the *Corridor* scene the disparity of the wall is not correct in *any* of the tested approaches. However, due to the cost function in the proposed approach, which allows pixels to be set to the disparity of the groundplane at a reduced cost, the proposed technique deals with the ambiguous disparity regions as designed, by setting them to the groundplane disparity. As previously stated, this is a design feature driven by the pedestrian detection module, as if ambiguous pixels are set to the groundplane disparity then they can have no influence, in either computational expense or algorithmically, upon the pedestrian detection technique. Similar examples exist in figures 4.30 and 4.31, where the disparity of the window (in the left of the *Grafton* scene) is consistently set to that of the groundplane.

With regards to the disparity of foreground pedestrians, the local based technique works well for medium distances, however the closer a pedestrian gets to the camera the more likely the local based technique will fail to assign the correct disparities due to the increase in size of homogeneous

image regions. As expected, due to the size of the aggregation window the disparity around depth discontinuities is blocky and lacks detail. The DP based technique suffers greatly from streaking effects and due to a lack of texture "holes" can appear in foreground objects, see figures 4.28 row 6, 4.30 row 5 and 4.31 rows 4 and 5. In contrast, the proposed technique appears robust with regard to both foreground and background regions, demonstrating strong inter-scanline consistency, good depth discontinues and a resistance to streaking effects. It also demonstrates the efficiency with which it can extract GCPs. This is shown by the fact that the dynamic disparity limit constraint is determined by the maximum GCP and background disparities. Therefore in order for the DP algorithm to assign the correct disparity of a foreground object it requires that a GCP of equal or greater disparity must exist in that (or previous) scanlines, i.e. if no GCPs are extracted in the scene then the disparity limit for a given scanline is set to the maximum background disparity for that scanline and so the correct disparity of a foreground object may not be assigned.

The proposed approach does have some drawbacks. For example, the strong inter-scanline consistency enforced by the vertical smoothness cost in the DP algorithm can result in a decrease in accuracy of the final disparities. An example of this can be seen in figure 4.29 row 5, where the top of a persons head is determined to be at a lower disparity due to a lack of texture and the disparity of previous scanlines. A similar example, but less severe can be seen in figure 4.27 row 7, where the lack of texture within the body means that, for a region, the cost of making a disparity jump down to the groundplane disparity is cheaper than the cost of staying at the same disparity and taking numerous smaller vertical smoothness costs. Finally, the vertical smoothness cost, coupled with the fact that the algorithm tends to assign ambiguous disparities to the groundplane can also introduce further artifacts in the disparity map if, for example, a homogeneous object appears only partly in the image and there are no GCP regions present to guide results. An example of this can be seen in figure 4.27 row 1, where due to the camera angle few GCP regions were created in either of the two people, therefore the disparity of their ambiguous pixels tended towards that of the groundplane for a number of lines. Once the disparity at these lines was initiated, due to the lack of texture it was always more expensive in each subsequent line to return to the correct disparity, and so apart from the pixels that were GCPs the disparity of the two people was set to that of the groundplane.



(a) (b) (c) (d)

Figure 4.27: *Overhead* scene results; (a) Input; (b) *SSD_sw35_t1000*; (c) *SSD_DP_o400_s400_t1000*; (d) Proposed approach.

(a) (b) (c) (d)

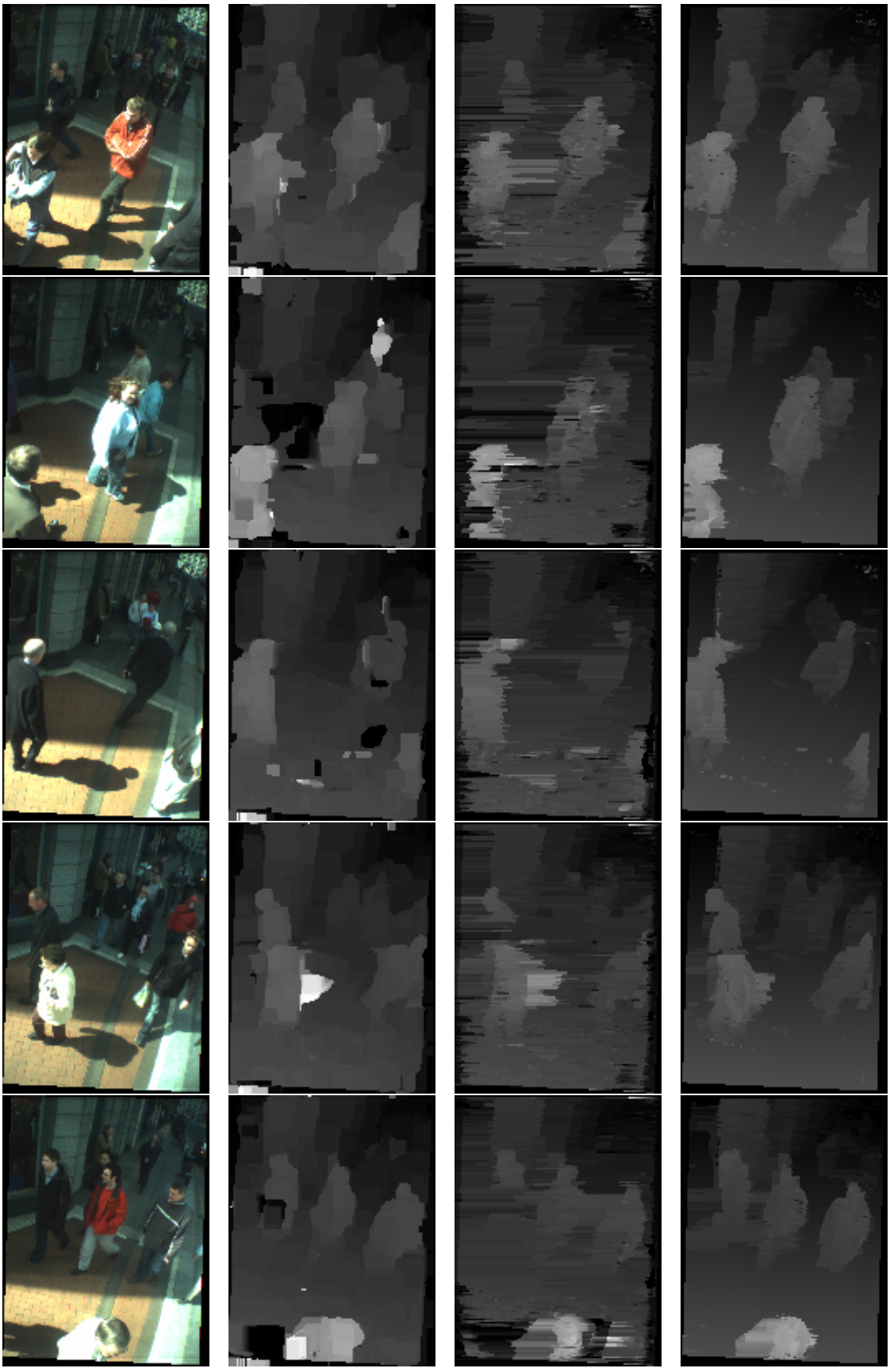
Figure 4.28: *Corridor* scene results; (a) Input; (b) *SSD_sw35_t1000*; (c) *SSD_DP_o400_s400_t1000*; (d) Proposed approach.

(a) (b) (c) (d)

Figure 4.29: *Vicon* scene results; (a) Input; (b) *SSD_sw35_t1000*; (c) *SSD_DP_o400_s400_t1000*; (d) Proposed approach.

(a) (b) (c) (d)

Figure 4.30: *Grafton* scene results 1; (a) Input; (b) *SSD_sw35_t1000*; (c) *SSD_DP_o400_s400_t1000*; (d) Proposed approach.



(a) (b) (c) (d)

Figure 4.31: *Grafton* scene results 2; (a) Input; (b) *SSD_sw35_t1000*; (c) *SSD_DP_o400_s400_t1000*; (d) Proposed approach.

4.5 Summary

In this chapter, disparity estimation techniques were reviewed and the first contribution of this thesis was presented. This contribution outlined a stereo correspondence technique, designed specifically for pedestrian surveillance type applications, which incorporates a number of novel enhancements to increase the robustness of the final disparity map. These enhancements include the use of *groundplane space*, the use of a *dynamic* disparity limit constraint, a technique to obtain Ground Control Points (GCPs) to help guide results, and a novel scanline cost function for the dynamic programming algorithm. The proposed technique was then quantitatively and visually evaluated against other standard techniques using both a synthetic dataset, designed to mimic typical pedestrianised scenarios and difficulties, and real-world data from a variety of indoor and outdoor scenarios involving a number of camera orientations, background scenes and lighting conditions.

The generation of a robust disparity map can be seen as part of the first module in the proposed pedestrian detection and tracking system, see figure 3.17. This disparity map, along with the background gradient model, is used as input to the pedestrian detection module, which is presented in the following chapter and is the second major contribution in this work. In this module, the disparity map is post-processed to remove any remaining artifacts and constrain the 3D points to a volume of interest. Thereafter the remaining disparity points are clustered into pedestrian regions via an iterative region growing framework that incorporates 3D information, groundplane estimation, non-quantised plan-view statistics and a basic human biometric model.

CHAPTER 5

Pedestrian Detection

5.1 Introduction

The first major contribution of this thesis was presented in chapter 4, where a stereo correspondence technique designed specifically for pedestrian surveillance type applications was presented. The generation of this disparity map can be seen as part of the first module in the proposed pedestrian detection and tracking system, see figure 5.1. This disparity map is post-processed and, along with the background gradient model, is used as input into the pedestrian detection module. This pedestrian detection technique, which is the second major contribution of this thesis, is presented in this chapter.

The proposed pedestrian detection technique clusters 3D points obtained from a post-processed disparity map via an iterative region growing framework that is robust to both under and over-segmentation. The technique is based on the use of 3D information, ground plane estimation and a basic human biometric model, which is incorporated directly into the region clustering algorithm. In addition, a novel plan-view statistic is applied to the final stage of clustering to increase robustness to both over and under-segmentation of pedestrians. Finally, the clustered regions are post-processed, using the background gradient model to aid in the removal of background regions and noise.

The proposed technique adheres to many of the requirements of a robust pedestrian detection technique, as outlined in section 1.1. In addition, it requires *no* external training, needing only to be calibrated with respect to the groundplane. However, the system does have a small number of drawbacks including; the camera must be orientated so that the groundplane is visible in the image plane, and the groundplane must be relatively flat with a relatively static background; the

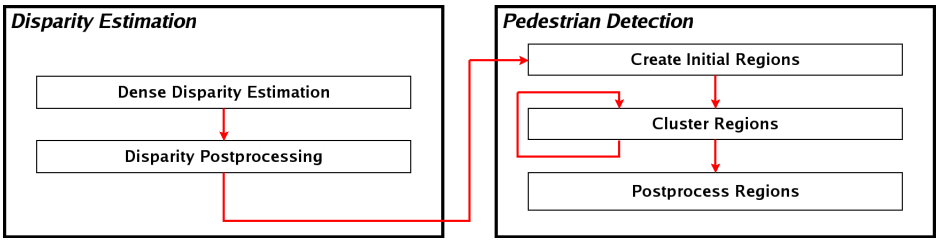


Figure 5.1: Basic system overview.

system is only able to reliably detect pedestrians for a short-medium range, up to a maximum distance of 8 metres from the camera; and the system assumes that a pedestrian is standing vertically with respect to the groundplane. The proposed pedestrian detection technique is quantitatively evaluated using a number of test scenarios in section 5.4. This evaluation is carried out using two differing methodologies; (1) using 2D techniques against a synthetic dataset and a number of real-world scenarios of varying camera height, camera orientation and environmental conditions; (2) using a 3D Vicon infrared motion analysis system [206] that provides an alternative evaluation methodology and also determines the accuracy of the proposed system with regard to 3D statistical data such as the height and positional deviation of detected pedestrians from their respective groundtruths.

This chapter is divided into four sections; section 5.2 outlines the post-processing techniques employed to remove disparity artifacts and constrain the input disparity map to those pixels that map to a 3D volume of interest; section 5.3 defines the proposed iterative region growing framework and post-processing steps applied to determine pedestrian regions from the post-processed disparity points; section 5.4 evaluates the proposed technique, using the two differing methodologies described previously, against a number of test sequences from a variety of scenes; and finally section 5.5 provides a discussion on the proposed approach, its evaluation and its effect on the choice of tracking methodology, which is presented in chapter 6.

5.2 Post-processing the Dense Disparity Map

The proposed pedestrian detection technique is based on an iterative region growing framework whereby the dense disparity map from the previous chapter is used as input. However, before this clustering is implemented the dense disparity map is post-processed for two reasons; (1) to constrain the disparity to pixels with belong to a 3D volume of interest (VOI), defined by a maximum and minimum height with respect to the groundplane in the scene and a maximum

distance from the camera; and (2) to remove artifacts generated during the disparity estimation process.

5.2.1 Defining a Volume of Interest

The first post-processing step involves removing disparity points that are outside a predefined volume of interest (VOI). In this work, the VOI is similar to that used in [11], whereby it is defined with respect to two planes; the camera plane and the 3D groundplane (which was introduced in section 4.3). Using triangulation and the input disparities, the 3D position of each pixel in I_1 can be calculated as long as a match for that pixel was found in I_2 – see section 3.2.2.3. In addition, using the 3D groundplane and equation 4.6 (see section 4.3.5) the height of each 3D point above the groundplane can be calculated in Euclidean co-ordinates. Therefore, each valid 2D point, u, in the input dense disparity image can be represented by its 3D co-ordinates with respect to a world Euclidean origin, $U = (x_u, y_u, z_u)$, and the Euclidean height of the 3D point above the groundplane, U_h.

A point is then defined as being outside the VOI and removed from the disparity image if any of the following is true:

1. No corresponding point to u could be found in I_2.
2. If $z_u > z_{max}$, where z_{max} is a threshold value that represents the maximum relevant distance from the camera along the camera's principal axis. In our experiments, z_{max} is set to 8 meters, due to the stereo camera rig baseline and the degradation of accurate stereo information beyond this distance (see section 5.3.3 for more information).
3. $u_h < h_{min}$. In our experiments, h_{min} is set to 0.9 meters ($\simeq$ 3 foot).
4. $u_h > h_{max}$. In our experiments, h_{max} is set to 2.1 meters ($\simeq$ 7 foot).

where h_{min} and h_{max} are set to the minimum and maximum expected pedestrian height above the groundplane, but allowances are made for fluctuations in the groundplane and disparity inaccuracies. After this step has been completed, the remaining disparity points are referred to as *foreground* disparities and the removed or invalid points as *background* disparities.

5.2.2 Removing Dense Disparity Artifacts

The second post-processing step involves filtering the remaining foreground disparities, removing possible artifacts that were generated during the disparity estimation process. As discussed in chapter 4, the disparity estimation technique is shown to produce robust results in a number of varying scenarios. However, the proposed disparity estimation technique can suffer, as can all other dense disparity estimation algorithms, from inaccuracies [207], especially in unconstrained conditions due to image noise, a lack of texture, occlusion or parameter choices in the chosen stereo correspondence algorithm. How these artifacts manifest themselves depends upon the stereo correspondence algorithm. In dynamic programming based algorithms they tend to take the form of horizontal streaks of inaccurate disparity data. A typical example of this can be seen in figure 5.2(c), which shows the foreground disparity of the pedestrian from figure 5.2(a). The lack texture within the midsection of the pedestrian in this figure is due to the colour of the pedestrians coat being similar to the background colour. After the first post-processing step, streaking can be seen to the left of the pedestrian's midsection. Also notice smaller streaks in the top left of the image. Fortunately, the streaks can be removed by searching the image for the characteristic vertical bar of constant disparity. It should be noted that this post-processing stage is not critical to the proposed pedestrian detection approach. It is, however, implemented to ensure that the final disparity map is as close a representation of the foreground regions of the scene as possible.

In order to remove streaks, the image is traversed vertically. When a pixel, (x, y), is a foreground disparity and the previous pixel, $(x, y-1)$, is not, the y value is noted as the start of a strip, y_{start}. The value of y is then incremented until (x, y) is a background point, whereupon the y value is noted as the end of the strip, y_{end}. The height of a strip is determined via $height_{strip} = y_{end} - y_{start}$. For a vertical strip to be part of a larger horizontal streak, then:

- $(x+1, y_{start}-1), (x+2, y_{start}-1)....(x+width_{strip}, y_{start}-1)$ must be a background disparity point, where $width_{strip} = height_{strip} * \mu$, where μ determines how wide a streak must be compared to its height for it to be retained. In our experiments $\mu = 1$.

- All points $(x+1, y), (x+2, y)....(x+width_{strip}, y)$ must be foreground disparity points and have the same disparity value as (x, y), where y is any value from y_{start} to y_{end}.

- $(x+1, y_{end}+1), (x+2, y_{end}+1)....(x+width_{strip}, y_{end}+1)$ must be background disparity points.

(a) (b) (c) (d) (e)

Figure 5.2: Remove streaks; (a) Input region; (b) Dense disparity; (c) Foreground disparity; (d) Streaks (in red); (e) Post-processed disparity.

If a streak is found, all points within it are declared as background disparity points. The value of $width_{strip}$ is incremented and the next column is tested. If the vertical strip continues onto this next column and the above 3 properties hold then the strip on this line is also removed and the process is reiterated. Figure 5.2(d) shows in red the regions that have been determined to be streaks in the foreground disparity image.

5.3 Proposed Pedestrian Detection Technique

To detect pedestrians, the resultant foreground disparities are clustered into coherent pedestrian regions via an iterative region growing framework. The clustering process is implemented in 3D, whereby the clustering of the regions is guided by a pedestrian model and 3D region statistics. As such, the technique is similar to some other 3D pedestrian detection techniques proposed in the literature, such as the subtractive clustering [11], the mean shift clustering [164], clustering via graphs [14], or clustering via plan-view statistics [166, 49, 21] (see section 3.4). However, the robust clustering of 3D points into distinct pedestrian objects poses many difficulties. Some of these include;

Disparity Map Quality As all disparity estimation techniques proposed in the literature can suffer from inaccuracies [207], the proposed clustering technique should therefore be robust up to some level of deterioration in the quality of the input disparity maps. If a simple region clustering technique is employed, such as [207], which clusters regions based upon the proximity of 3D points and some predefined heuristics, then such deteriorations in quality may result in over-segmentation (where a single person is split into two or more regions) or under-segmentation (where a two or more people are clustered into a single region) of pedestrians. The first scenario

could occur if the disparity map is overly noisy, which may result in a person being segmented as one or more distinct regions of constant disparity. Alternatively, if the disparity map is overly smooth, whereby smooth disparity discontinuities are created at object boundaries, then under-segmentation is possible. Figure 5.3 illustrates this point. The foreground disparities of figure 5.3(c) are triangulated and presented in a plan-view orientation in figure 5.3(d) (this transformation is solely for illustrative purposes). To guide visualisation, the majority of the points in the red box of figure 5.3(d) belong to person **A** in figure 5.3(a). In order to segment person **A**, all the 3D points belonging to this pedestrian should be clustered together into one distinct region, as shown in figure 5.3(g). However, due to the smoothing effect, the disparities may change gradually from one disparity level to another at object boundaries, see figure 5.3(h) where the smoothed points are highlighted in red. Due to these smoothed points, under-segmentation may occur if there are a number of smoothed points between two distinct objects and the jump in disparity between each smoothed point is below the required threshold needed to merge regions together.

Pedestrian Model In order to limit possible under-segmentation, many techniques incorporate a pedestrian model into the clustering framework that limits clustered regions to maximum sizes. However, the 3D clustering techniques that adopt such models must ensure that they are correct in size. If they are too small, then over-segmentation can occur as not all 3D points belonging to a distinct person can be clustered into one coherent region, causing the detection of false-positives. If they are too large, then under-segmentation is likely when pedestrians come close together due to 3D points from other objects not belonging to a single pedestrian becoming clustered into a region. A constant size model, however, is difficult to define as different people, including children and adults, and different poses, have different associated heights and widths.

If the correct pedestrian model is selected, then it is possible to correctly cluster the 3D points belonging to an individual person, see figure 5.3(i). However, the positioning of this model is crucial. For example, in figure 5.3(j) the correct pedestrian model is incorrectly positioned in 3D, resulting in a second cluster forming from the 3D pixels of person **A** that are outside the range of the first pedestrian model. The result of this incorrect positioning is the over-segmentation of pedestrian **A**. The correct position of this model is difficult to obtain. In general, it is not computationally possible to obtain a globally minimum solution via a hypothesis-and-test approach, especially when the number of pedestrians and their relative scales is also unknown.

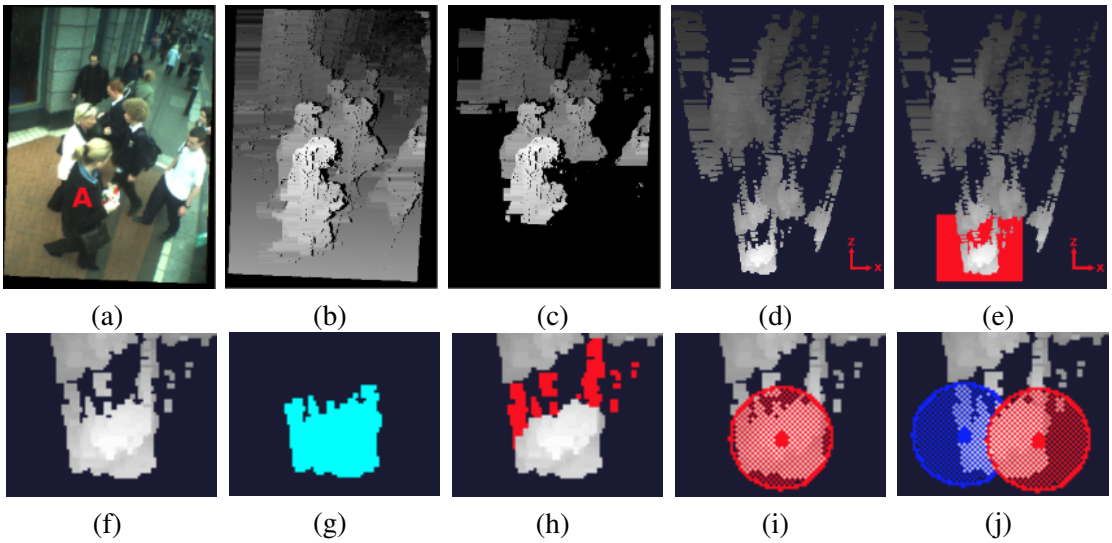


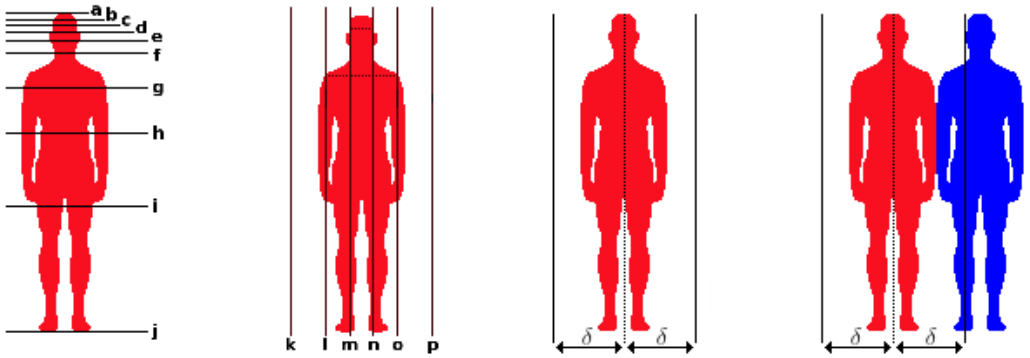
Figure 5.3: Region clustering; (a) Input scene; (b) Dense disparity map; (c) Foreground disparities; (d) Plan-view foreground disparities; (e) Plan-view foreground disparities section; (f) Plan-view foreground disparities; (g) Required final cluster; (h) Smoothed points; (i) Correct cluster centre; (j) Incorrect cluster centre.

Background Regions Not all 3D points in the scene will belong to pedestrian objects, therefore background regions should be removed in a robust manner that is relatively invariant to real-world environmental conditions, such as rapidly changing lighting conditions. In addition, it should be noted that most techniques, including the one proposed here, makes the assumption that all other objects that are not due to background objects are pedestrians and therefore should be clustered as such. Due to the range of the camera, and the envisioned applications, this is not an unreasonable assumption in crowded situations if the likelihood of all objects in the scene being pedestrian is high, e.g. in pedestrianised urban areas.

The proposed technique differs from other 3D pedestrian detection clustering techniques proposed in the literature. These enhancements are made to reduce, or remove, some of the trade-offs involved in the selection of a pedestrian model, to select the best position for each model centre (without prior knowledge of the number of pedestrians in the scene) and to increase robustness to real-world conditions. The first contribution in this regard is the choice of the model chosen to represent pedestrians within the clustering framework.

5.3.1 Biometric Model

As outlined in the previous section, the choice of a constant size model to cover all types of pedestrians, including children and adults, is problematic. If they are too small or large, then over



(a) (b) (c) (d)

Figure 5.4: Golden ratio; (a) Vertical; (b) Horizontal; (c) δ should be large enough to avoid over-segmentation; (d) δ should be small enough to avoid under-segmentation.

Distance	Meaning
|aj|	the height of the human body
|ac|	the distance from the top of the head to the forehead
|ad|	the distance from the top of the head to the eyes
|mn|	the width of the head
|af|	the distance from the top of the head to the base of the skull
|lo|	the width of the shoulders
|ah|	the distance from the top of the head to the navel and the elbows

Table 5.1: Biometric distances overview.

and under-segmentation can respectively occur. In the proposed technique, a simple biometric person model is defined that is dependent on the position of the groundplane in the scene, the 3D points that constitute clustered regions and the *Golden Ratio*. As such, once the groundplane is calibrated with respect the camera rig the model requires *no* external training.

The pedestrian model is based on $\Phi = \sqrt{5} * 0.5 + 0.5 \simeq 1.618$. This number is known as the *Divine Proportion* or the *Golden Section/Ratio/Mean/Number* [208] and can be applied to define the proportions of a human body. Figure 5.4(a) illustrates how a body is segmented using Φ. Let |aj| be the Euclidean distance between the horizontal lines a and j. Therefore, |aj| is the height of a human body. Using Φ and |aj|, various other proportions of the human body can be defined.

In figure 5.4(a) $|ai| = \frac{|aj|}{\Phi}$, $|ah| = \frac{|ai|}{\Phi}$...$|ab| = \frac{|ac|}{\Phi}$ [208]. In addition to is segmenting the body vertically, the golden ratio can also be employed to define the *width* proportions of a human body. For example, in figure 5.4(b) |mn| is equivalent to |ae|. Similarly $|lo| \equiv |ag|$ and $|kp| \equiv |ah|$. Distances of interest in the proposed approach are outlined in table 5.1.

This biometric information constitutes useful information for assisting region clustering. Statistics, such as a region's maximum height above the groundplane, can be built up for each region as they are created. Using these statistics, and the assumption that the region belongs to a pedestrian,

the value of |aj| can be equated to the maximum region height. Therefore, using Φ in conjunction with |aj| other metrics such as the width of a pedestrian's shoulders, |lo|, can be determined. This value of |lo| is dependent on the height of a region above the ground, therefore the width of a child's shoulders is less than that of an adult's shoulders. Using this information, a maximum cluster size can be dynamically determined for each pedestrian during the clustering process.

However, in order to cluster a single pedestrian into a distinct region, the pedestrian model, its position and orientation in 3D space are required. By making the assumption that people in the scene are standing upright, the orientation of the model can be set to be parallel to the 3D groundplane normal. In addition, as illustrated in figure 5.3, the centre of the pedestrian model should be positioned correctly with respect to the pedestrian's body. Unlike previous techniques, such as [21], an assumption is not made that the best position for the model is that of the maximum peak height of a person above the groundplane, as this is likely to be perturbed by noise and pedestrian pose. Instead, the model is positioned such that the centre of the model passes through the centre of mass of a pedestrian region as is less affected by either these phenomena. Therefore, the pedestrian model is positioned with respect to a 3D line, or axis, which is parallel to the 3D groundplane normal and passes through a regions centre of mass. Let this 3D line be called the *central axis* of the region.

Therefore, by obtaining the 3D statistics of a region, which includes its maximum height and its centre of mass, a pedestrian model can be scaled and positioned with respect to the region. Once this model is positioned, all 3D points within a certain distance of this central axis may then be considered to be part of the region. This idea can be represented in 3D by a cylinder that is centred on the pedestrian's central axis and has a radius defined by δ, which is some function of |aj| and Φ. Any 3D point that is inside of this cylinder is a point that belongs to the pedestrian. This value of δ should be large enough so that if the model is positioned correctly, the pedestrian region does not become over-segmented for a variety of poses and orientations, see figure 5.4(c). The value of δ, however, should be small enough so only pixels from one pedestrian should be clustered together, see figure 5.4(d). In general, there is a trade-off between the value of δ and possible over and under-segmentation. In the proposed approach, the maximum value of δ is set to |lo|, or the width of a pedestrians shoulders, as illustrated to scale in figures 5.4(c) and (d). The iterative nature of the algorithm and a novel plan-view statistic are used to prevent under-segmentation of pedestrians.

5.3.2 Region Clustering

An overview of the pedestrian detection algorithm is presented in figure 5.5. In addition, an illustrative example of the clustering process involving two pedestrians is given in figure 5.6. Initially foreground disparities are obtained using the post-processing disparity techniques previously outlined. These disparity points are triangulated to obtain their respective 3D co-ordinates, illustrated in figure 5.6(e). Note that for ease of illustration, figures 5.6(e)-(n) are depicted from a plan-view orientation, whereby the viewing angle is parallel with the groundplane normal and the 3D points are orthographically projected onto a 2D plane. It must be stressed however, that this is for illustrative purposes only and the clustering process groups 3D points, not 2D points.

The clustering of the foreground 3D points is a two stage process. The first stage of clustering is implemented using a framework that is highly constrained, therefore only very small clusters are allowed to form, see figure 5.6(g). This is achieved by setting the biometric threshold value, δ, to a small value, see figure 5.5. This initial stage is implemented to both reduce computational complexity, and to obtain more accurate region statistics before the second stage. The second stage is implemented in an iterative clustering framework, see figure 5.5, whereby each stage increases the biometric threshold value, δ, thereby allowing regions of greater distances to be clustered together, see figures 5.6(h)-(j). During each iteration, the 3D statistics of each region, such as the position of central axis and the maximal region height, are constantly recalculated to ensure that the pedestrian models are positioned and scaled correctly.

The final stage of clustering, when δ is at its maximum value, incorporates a novel plan-view statistic designed to increase robustness to both over and under-segmentation of pedestrians. This plan-view statistic differs from those defined in [166, 49, 21] (see section 3.4.1) as there is no quantisation resolution required for the groundplane. This is a major advantage over other plan-view statistic techniques as the inherent trade-off between over and under-segmentation in the resolution is eliminated.

5.3.2.1 Stage 1: Creating Initial Clusters

The first stage of the proposed algorithm clusters points via an 8 neighbourhood connected components algorithm. The image is traversed from left to right in a top-down manner. If a 2D foreground disparity pixel, u, cannot be merged to any neighbouring regions, it is initialised as a new 3D region using its triangulated 3D point, U. The height above the groundplane of this

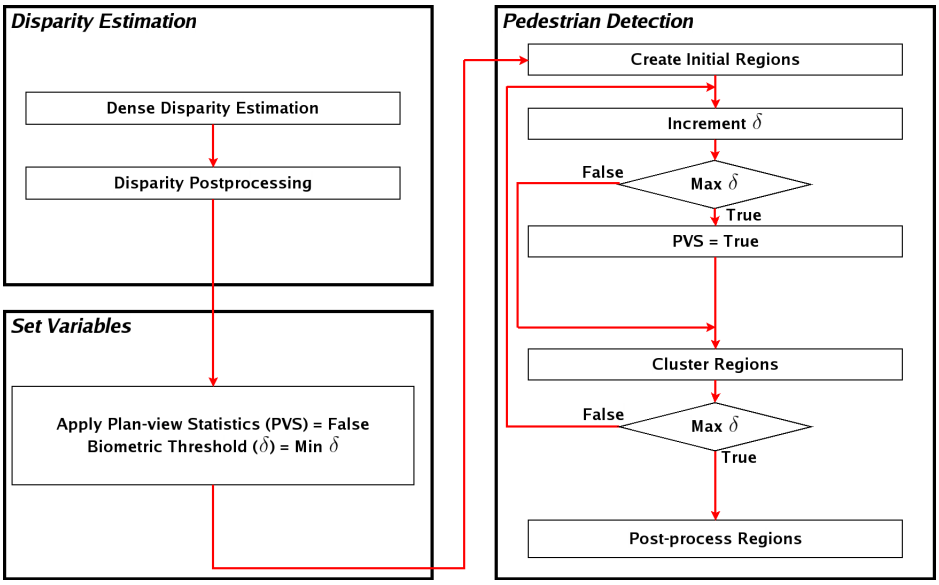


Figure 5.5: Detailed system overview.

point, U_h, is stored as the regions maximum height, reg_h, and the regions central axis, reg_{cx}, is set through U parallel to the groundplane normal.

To test if the pixel, u, is allowed to merge with a previously initialised region, reg, the maximum height between U and reg is initially obtained. This height value is used to define a 3D distance value, δ, using the biometric pedestrian model. U is allowed to merge with reg if

- in 2D, u neighbours a pixel, v, of *identical* disparity that is a projection of a 3D point V in reg
- in 3D, $dist(U, V) \leq \delta$
- in 3D, $dist(U, reg_{cx}) \leq \delta$

where $dist$ is the Euclidean distance. If u is allowed to merge with reg then the 3D statistics reg_h and reg_{cx} are updated as necessary, where reg_{cx} passes through the centre of mass of the region.

To test if the region, reg^1, is allowed to merge with a second region, reg^2, the maximum height contained in both reg^1 and reg^2 is initially obtained. This height value is used to define δ. reg^1 is allowed to merge with reg^2 if

- in 2D, two neighbouring pixels, u and v, are projections of 3D points, U and V, in reg^1 and reg^2 respectively
- in 3D, $dist(U, V) \leq \delta$

- in 3D, $dist(U, reg^2_{cx}) \leq \delta$, or vice versa
- in 3D, $dist(reg^1_{cx}, reg^2_{cx}) \leq \delta$

If reg^1 is allowed to merge with reg^2 then the 3D statistics of reg_h and reg_{cx} are updated as necessary.

In the first stage of the clustering process $\delta = |\text{ad}| * reg_h$, where the height of a region is defined by reg_h. This initialises δ as a value of roughly 0.05% of the height of the region. This small distance is intended to create a large number of small regions, see figure 5.6(g), where each colour represents a distinct region. Figure 5.6(f) illustrates the heights of 3D points above the groundplane, where the brighter the colour, the greater the height. Clusters are initialised in this way to avoid two areas of separate objects, which are separated by a small Euclidean distance in 3D, to become merged.

5.3.2.2 Stage 2: Iterative Region Growing

Throughout the clustering process δ is increased gradually from |ad| to the maximum value of $\delta = |\text{lo}|$. In this way, each separate object region can be allowed to grow in isolation and avoid being merged. It allows the merging of regions if $dist(reg^1_{cx}, reg^2_{cx}) \leq \delta$, as described in stage one. In this way two regions can be merged even if they do not appear directly beside each other in 2D. Instead, the orthographic projection onto the groundplane for each point in each region is then obtained using equation 4.6 (see section 4.3.5). This path is traversed, and any two regions can be tested and possibly merged if they both appear on that path.

The iteration from $\delta = |\text{ad}|$ to $\delta = |\text{lo}|$ should ideally be made made as slowly as possible. If the transition from |ad| to |lo| is made too quickly, then the regions may be grown too quickly, resulting in the incorrect positioning of pedestrian models leading to poor segmentation of the final pedestrian regions. However, to reduce computational complexity the number of iterations should be kept relatively low. In our experiments, the clustering is implemented in seven distinct steps. Seven was chosen since using $0.5 \times \Phi$ there are 7 steps to go from |ad| to |lo|; this leads to a reasonable trade off between complexity and robustness to segmentation problems. Figures 5.6(g)-(j) depict the regions at various stages of the process. Notice that due to the limited value of δ, regions from different objects are grown in isolation, thereby eliminating under-segmentation up to this point.

However, for the final iteration, when the maximum value of δ is set, this technique alone

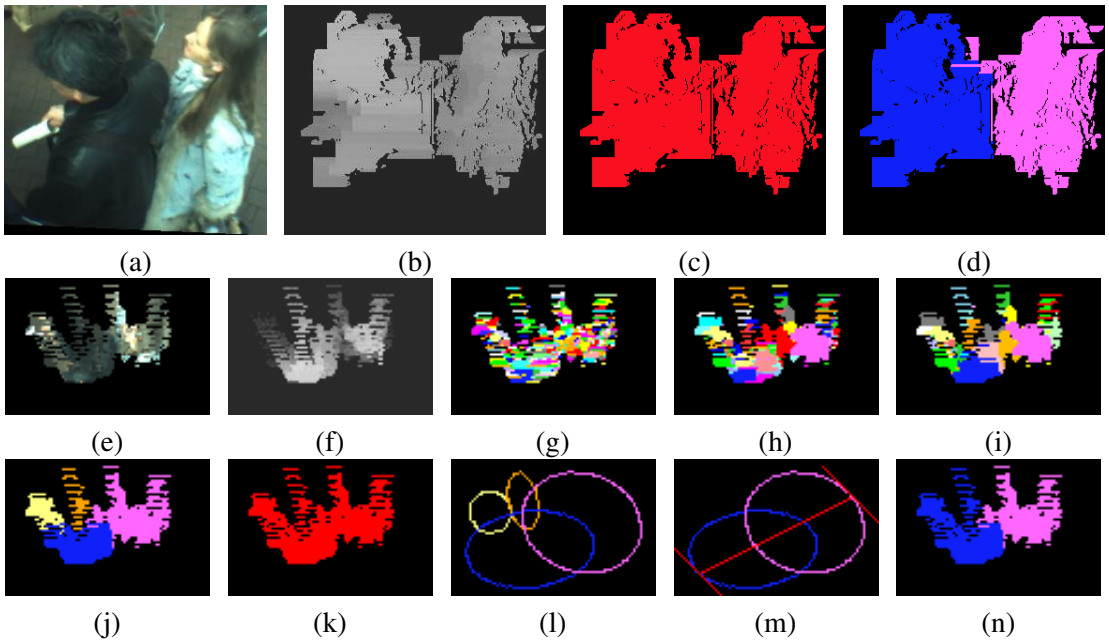


Figure 5.6: (a) Image tile showing two pedestrians very close together; (b) Foreground disparity; (c) & (k) Final regions (i.e. 7^{th} iteration) without the under-segmentation test; (d) & (n) Final regions (i.e. 7^{th} iteration) with the under-segmentation test; (e) 3D points from a plan-view orientation; (f) 3D point heights; (g) Initial regions; (h) Region clustering - 1^{st} iteration; (i) 2^{nd} iteration; (j) 6^{th} iteration; (l) Best-fit ellipses of the 4 regions from (j); (m) Region diameter.

is prone to under-segmentation if two pedestrians are positioned very close together, see figures 5.6(c) and (k). In this example, under-segmentation occurred as the Euclidean distance between reg^1_{cx} and reg^2_{cx} is slightly less than the maximum value of $\delta = |\text{lo}|$. In addition, if the distance between two other regions that belong to a single pedestrian is slightly greater than $|\text{lo}|$, then they are not merged and thus over-segmentation occurs. The technique is therefore prone to both over and under-segmentation as the clustering technique is based *solely* on the position of the central axes of regions, without taking into account the global features associated with the regions.

Robustness to Under- and Over-segmentation A contribution of this thesis is augment the clustering framework described previously with a novel plan-view statistic that incorporates the required global feature information from regions. This in turn leads to increased robustness to both under- and over-segmentation of pedestrians via the proposed approach. The creation of this plan-view statistic and its use for a reduction in both under- and over-segmentation will now be examined.

During the final iteration of the clustering algorithm (i.e. when δ is at its maximum value of $|\text{lo}|$), robustness to under-segmentation can be enhanced by invoking an additional constraint on the clustering of two regions. This additional constraint, which is referred to as the *under-*

segmentation test, is designed to compare the global shape of the two regions to determine the possible presence of two people. The under-segmentation test incorporates a novel plan-view statistic that approximates the global shape of each region by a *best-fit ellipse* around the shoulder height of each region (see section 5.3.2.2 for details on how to obtain this ellipse). Figure 5.6(l) illustrates the best fit ellipses for each of the four regions of figure 5.6(j) that exist before the final clustering iteration occurs. Using these region statistics, two regions, reg^1 and reg^2, can be merged if two constraints are passed

1. $dist(reg^1_{cx}, reg^2_{cx}) < |\text{lo}|$, which states that two regions can only join if the distance between their centres is less than the shoulder width of a person, and

2. $\gamma < 2\,|\text{lo}|$, which is defined as the *under-segmentation test*. In this inequality, let γ be the maximum Euclidean distance between two region ellipse points on the line l, where l is a 2D line that passes through the centre of the two region ellipses – see figure 5.6(m).

The *under-segmentation test* ensures that for two regions to be merged, the distance across the two regions, parallel to their centres, must be less than the combined shoulder widths of two people. This constraint, in addition to the first central axis constraint, creates a powerful pair of clustering constraints that result in a significant reduction in under-segmentation.

Robustness with respect to over-segmentation can also be enhanced using similar techniques via an *over-segmentation test* that is implemented in the post-processing stage of region clustering. For example, if from two regions, reg^1 and reg^2, the statistics show that $dist(reg^1_{cx}, reg^2_{cx}) > |\text{lo}|$ but $\gamma < 2\,|\text{lo}|$ – then the two regions may belong to *either* 1 or 2 people (if it is the latter, then merging the two regions would result in under-segmentation). In order to determine whether the two regions should be merged, further examination of the regions is required. In the proposed approach, the two regions are allowed to merge if the diameter of a second best-fit ellipse, fit to only the 3D points located above shoulder height, equates to the size of a single person's head. Using this approach, two best-fit ellipses (one from each region) are obtained, using Φ to constrains the 3D points used in the creation of the ellipse to those above neck height (i.e. higher than $|\text{aj}| - |\text{af}|$). If the radius of the major axis of both the ellipses are greater than half the width of a head, $\frac{|\text{mn}|}{2}$, then it is determined that two head regions do indeed exist and therefore merging *cannot* occur. Otherwise, the merging of reg^1 and reg^2 is permitted. Therefore, using this technique, reg^1 and reg^2, can be merged if

1. $dist(reg^1_{cx}, reg^2_{cx}) > |\text{lo}|$, and

2. $\gamma < 2\,|\text{lo}|$, and

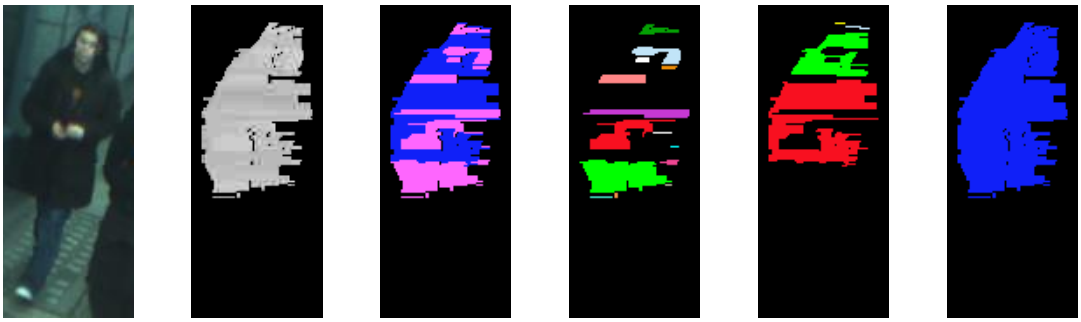
3. $\alpha < \frac{|\text{mn}|}{2}$ and $\beta < \frac{|\text{mn}|}{2}$, where α and β are the major axis diameter of the "head region" ellipses from reg^1 and reg^2 respectively.

Best-Fit Ellipse To determine the best-fit ellipse of a region, a 3D point set is created using Φ to obtain all 3D points in the region that are at or above a particular height – shoulder height is chosen in the *under-segmentation test* as the best area to fit the ellipse, rather than the region as a whole, as this area is less likely to be perturbed by objects, such as backpacks or outstretched limbs. These 3D points are then orthographically projected onto the groundplane, which removes one degree of freedom from the points. This is similar to the techniques used in the generation of other plan-view statistics such as those used in [105, 21, 209, 122] except that the points are *not* quantised into discrete bins. The best-fit ellipse is then obtained from the resultant 2D point set in a manner similar to that presented in [210].

5.3.3 Dealing with Distant Pedestrians

A prerequisite of the proposed algorithm is good disparity estimation and 3D reconstruction. The more accurate these are, the better the subsequent segmentation. As described in chapter 4, various techniques are employed to ensure robust disparity estimation. However, most stereo correspondence algorithms (including the one employed by the authors) compute the disparity of a given point to be a discrete value between 0 to n, where n is defined by the disparity limit constraint. This means that if the disparity changes within an object then the disparity difference has to be ≥ 1. When the object is close to the camera, a change in disparity of 1 between two pixels, u and v, still results in a smooth surface as the Euclidean distance between the 3D position of the points, U and V, is relatively small. However, the farther away an object becomes from the camera, the greater an effect a change of disparity will have in terms of Euclidean distance. For example, if the disparity values at u and v were 1 and 0 respectively, then the Euclidean distance from U to V becomes ∞.

Therefore, the farther away the pedestrian is from the camera, the more likely it becomes that the 3D points belonging to a single person become spread out [211]. In addition, there are fewer 3D points belonging to the pedestrian and therefore the central axis of a clustered region becomes more susceptible to noise. A repercussion of this is that as the distance of a pedestrian from the



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

Figure 5.7: Characteristic splintering observed for distant pedestrians; (a) Image data; (b) Foreground disparity; (c) Over-segmented region; (d) reg_1 sub-regions; (e) reg_2 sub-regions; (f) Merged region.

camera increases, then the likelihood of the two regions, reg_1 and reg_2, belonging to the same pedestrian having either $\gamma > 2\,|\text{lo}|$ or $d_{cx}^{12} > |\text{lo}|$ increases. This can result in over-segmentation of a pedestrian, as seen in figure 5.7(c).

Solutions to this problem include to; (1) turn off the under-segmentation test for regions at distances greater than a certain distance, $dist_z$; (2) allow an increase in the value for $|\text{lo}|$ for regions at distances greater than $dist_z$; or (3) take into account the characteristic appearance of distant over-segmented regions, and merge them appropriately. The first two options both involve an unknown threshold, $dist_z$, and both are subject to causing unnecessary under-segmentation. In this thesis, the third solution is adopted and it is observed that, in general, over-segmentation at large distances results in a characteristic *splintering* of regions in 2D image space. In the proposed approach, this splintering is defined to have occurred if; (a) each of the two regions, reg_1 and reg_2, are composed of more than one disjointed sub-regions in *2D* image space – see figures 5.7(d) and (e) where each sub-region of each of the two regions of figure 5.7(c) is coloured differently; and (b) the merging of reg_1 and reg_2 would result in two or more of the sub-regions in *each* of reg_1 and reg_2 becoming connected in 2D image space – see figure 5.7(f) where all the sub-regions of figures 5.7(d) and (e) are now connected in 2D image space. Using this proposed approach, if two regions, reg_1 and reg_2, are found to be splintered, then the under-segmentation test for the regions is not employed and reg_1 and reg_2 can be merged simply if $d_{cx}^{12} < |\text{lo}|$. It has been found that this splintering test works as well as either option (1) or (2), but without the need to set any external thresholds.

5.3.4 Region Post-processing

Using the proposed approach, 3D points belonging to both foreground and background objects within the volume of interest are clustered together into pedestrian shaped regions without distinction, see figure 5.8(c), where each colour depicts a separate region. The final stage in the pedestrian detection module therefore involves post-processing to remove regions (or parts of regions) caused by noise and background objects. However, this post-processing stage must be implemented in a manner that is robust to changes in real-world conditions, such as lighting conditions. The post-processing for the proposed technique can be sub-divided into three main stages; (1) each object is trimmed of any background pixels; (2) objects caused by noise are removed; and (3) pedestrians are segmented from background objects, such as walls. Each stage is guided by the background gradient model, first introduced in section 4.3.6.1.

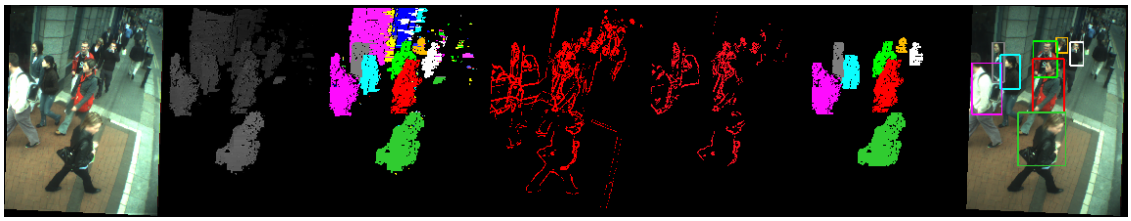
Using the background gradient model, foreground gradients can be obtained using background subtraction techniques, see figure 5.8(d). However, this foreground is still highly susceptible to noise and illumination changes at strong background gradients, for example, in figure 5.8(d) some areas of the groundplane and background wall have been labelled as "foreground". The foreground gradient regions are therefore post-processed via three techniques; (1) foreground gradients that do not have a corresponding disparity within the required VOI are removed, thereby eliminating unrequired data; (2) foreground gradients that have *similar* angles to their corresponding background gradient angles are removed (in our experiments the threshold is set to 15 degrees), thereby eliminating gradients that may have been caused due to changes in lighting conditions that may have significantly increased or decreased gradient strength; (3) the remaining foreground gradients are clustered together using a basic 8-neighbourhood connected components algorithm. During this clustering process the height of each foreground gradient pixel is obtained and the maximum and minimum height statistics of each region is obtained, each of the final gradient clusters must span height range, min_h_{grad}, which is set to 10cm in our experiments. If a region does not span this range, then it is considered as noise and removed. The final post-processed gradients can be viewed in figure 5.8(d). It is acknowledged that many legitimate foreground gradient points have been incorrectly removed during this post-processing, however as the foreground gradients are used for *guidance* in the segmentation of foreground objects, the loss of some of these points is acceptable and does not greatly affect the overall approach. For the remainder of this section post-processed foreground gradients are referred to as simply FGs.

The first post-processing stage trims each of the clustered regions of any background pixels. This is simply achieved by applying the use of FGs and resizing the bounding box of each region such that each edge of the box passes through a FG contained within the region. After this stage, all points that occur outside the bounding box are discarded. If there are no FG pixels inside a region, then that region is simply removed.

The second stage removes regions that are most likely caused by noise. This is achieved by; (1) removing all regions that consist of a total number of pixels less than a threshold, $pixels_{min}$, which is set to 500 in our experiments, assuming this to be the minimum size of a pedestrian in pixels; (2) in a similar process to post-processing FGs, as the height of a pixel is known, if the absolute difference between the maximum and minimum height values appearing in a region is less than a threshold, min_h_{reg}, then the region is also removed. In our experiments min_h_{reg} is set to 20cm.

The final post-processing step is implemented for two reasons, the first is to remove any remaining background objects that has passed the first two sections, possibly due to image noise causing some false-positive FGs to remain, the second is to segment pedestrians from high background objects, such as walls. The latter scenario can occur if pedestrians are positioned at relatively the same depth as background objects, such as walls. Usually, a single pedestrian is segmented from the wall using the first post-processing technique, however difficulties can arise if two or more pedestrians in close proximity are positioned close to a *high* wall and the VOI is set such that the high pixels are within the VOI. Due to the algorithm's clustering technique and the biometric model, these high points would ensure a large maximal Euclidean value for δ, thereby possibly clustering a number of pedestrians in the same region. To separate out these foreground objects each pixel in a region is classified as either; (a) foreground, if the pixel is a FG *or* if the background gradient is strong (i.e. over the $min_{grad} = 10$ threshold of section 4.3.6.1) and the current pixel is not a strong gradient; otherwise (b) background, if the current pixel is a strong gradient; otherwise (c) unknown. Using these classifications the total percentage of background to foreground pixels is determined within a region. If this percentage is greater than 20%, then the biggest gap, horizontal to the groundplane, between two FGs within the region is obtained and removed. If this technique results in the region being split into two, then two new regions are created. This process is continued until the total percentage of background edges within each region is less than 20%.

Using this technique of background subtraction greatly reduces errors in real-world environ-



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

Figure 5.8: Post-processing clustered regions; (a) Input image; (b) Foreground disparity; (c) Clustered disparity regions; (d) Foreground gradients; (e) Post-processed gradients; (f) Post-processed regions; (g) Final pedestrian bounding boxes.

ments as there is less reliance on the background subtraction algorithm. For techniques such as [106, 4, 56], etc, pedestrians are detected from the foreground subtracted pixels and therefore the success of the technique is limited by that of the background modelling technique. The proposed technique, however, creates objects and removes those caused by the background using a *subset* of the strongest sparse foreground gradient regions. Using this technique the reliance upon the background model is reduced significantly. The robustness of this technique of background subtraction, and the proposed pedestrian detection as a whole, is examined in the following section.

5.4 Experimental Results

We quantitatively evaluated the proposed pedestrian detection technique using two differing methodologies. The first technique, presented in section 5.4.1, evaluates the proposed approach via 2D image plane comparison techniques. In each input frame a number of manually annotated pedestrian bounding boxes are created. These groundtruth bounding boxes are then compared to the final pedestrian regions obtained using the proposed approach. If the overlap between a groundtruth and a detected region's bounding box is large enough then a match is declared. Using this technique *precision* and *recall* values are obtained for the proposed technique in a number of scenarios.

The second technique, described in section 5.4.2, evaluates the proposed pedestrian detection technique via 3D groundtruth information. Using this technique, the groundtruth 3D position of each pedestrian in a given image is obtained. This information is then compared to the final 3D positions of pedestrians obtained using the proposed approach. If the Euclidean distance between a groundtruth and detected pedestrian's 3D position is less than a threshold, then a match is declared. In addition to precision and recall values, further evaluation metrics such as the average error in the obtained 3D pedestrian statistics are calculated. Finally it should be noted that using both methodologies only one match between a detected pedestrian and a groundtruth pedestrian is

permitted.

5.4.1 2D Evaluation

The first technique applied to evaluate the proposed approach is based on traditional 2D image plane comparison techniques. In this approach, the synthetic data of section 4.4.2 plus five real-world test sequences were manually groundtruthed by positioning a separate bounding box around each person in an image. A correctly segmented pedestrian is then defined as a region that overlaps a groundtruth area by a predefined threshold.

5.4.1.1 Dataset Sequences

For this evaluation three different real-world test scenarios were identified with varying camera height, camera orientation and environmental conditions; the *Overhead*, *Corridor* and *Grafton* scenarios, introduced in section 4.4.3.2. From these three scenarios 5 test sequences of resolution 640×480 were captured between 2–6.5Hz [1]. As previously stated, these experimental sequences were chosen to test the proposed technique extensively in several areas, such as disparity estimation, foreground segmentation, pedestrian detection and tracking, and none of the test sequences were used in development of the proposed algorithms. In addition, for each of the sequences no restrictions or instructions were provided as to where people could go, what they could do or what they could wear.

The five sequences consist of 1 from the *Overhead* scenario, 1 from the *Corridor* scenario and 3 from the *Grafton* scenario. An overview of these sequences is provided in table 5.2. In addition to varying camera orientations and positions, the sequences exhibit a variety of illumination conditions. The *Overhead* sequence consists of stable lighting conditions, with brightly illuminated with a highly reflective ground surface. The *Corridor* sequence is brightly illuminated on one side, and dark on the other side, due to skylights in the corridor. These skylights also cause fluctuations in the ambient lighting of the scene due to varying cloud coverage. The first two *Grafton* sequences exhibit relatively constant illumination conditions that minimise shadows cast and background illumination changes. However, the illumination conditions in *Grafton* sequence 3 are constantly changing. It consists of 3 vastly differing lighting conditions in its 63 second

[1] It should be noted that these are the *average* frame rates. In all the test sequences, the latency between frames varied due to the software employed to stream the images from the Digiclops camera to a computer hard disk. In the evaluation datasets, the minimum and maximum latency between frames were recorded at 0.08 and 1.5 seconds respectively.

Sequence	*Num. Pedestrians*	*Num. Frames*	*Hz*	*Time (in Minutes)*
Grafton 1	666	124	≈ 2.0	1.03
Grafton 2	754	106	≈ 2.0	0.88
Grafton 3	555	127	≈ 2.0	1.06
Grafton Total	1975	357	≈ 2.0	2.97
Overhead Total	657	418	≈ 6.5	1.10
Corridor Total	1027	697	≈ 5.3	2.26
Total	3659	1472	≈ 2.0–6.5	6.33

Table 5.2: 2D dataset sequences.

duration. To illustrate the severity of these conditions, the input frames of figure 2.1(a)-(g) (see section 2.2) are taken from within *Grafton* sequence 3, and the illumination change within these frames occurs in 7 seconds.

5.4.1.2 Evaluation

In this evaluation process, a person is defined as someone who has a section of their body above the waist, no matter how small, visible in the image. If all that can be seen of a person in the image is an outstretched hand or a backpack then they *are* counted as being present. However, if just a leg or foot is present, then they are *not* counted. It should be noted however that due to the camera offsets in the stereo rig not all pedestrians visible in one camera are visible in the other (especially at the edges of the images). In this approach the right camera of the stereo rig is used, and therefore all groundtruths are created with respect to this camera regardless of whether or not the pedestrian appears in the alternative image. The only other constraint for creating groundtruth regions was that people who are further than 8 metres from the camera are *not* considered valid pedestrians. This constraint is necessary in some sequences as pedestrians can be seen for over a hundred metres, placing bounding boxes around all of these people and to groundtruth against them would introduce a lot of noise into the evaluation. The distance of 8 metres was chosen as the cutoff point as the proposed pedestrian detection system removes all 3D points greater than this distance from the camera as the disparity map quality degrades rapidly after this point as described in section 5.2.1.

Using this technique, the proposed algorithm was evaluated against the manually annotated groundtruth for the 5 test sequences and also on the synthetic data of section 4.4.2. The results of this evaluation can be seen in tables 5.3 and 5.4, where; *Groundtruth*, represents the number of people present in the groundtruth data; *Detected*, represents the number of distinct regions the pro-

posed pedestrian detection algorithm detected in the sequence and; *Correct*, represents the number of *Detected* regions that correctly overlapped with the *Groundtruth*. A correctly segmented pedestrian is defined as a region that overlaps a groundtruth area by 50% or more. It is acknowledge that this percentage is relatively low, but this work is more interested in detecting pedestrians than detecting the correct number of pixels corresponding to a person. As such, this percentage threshold was chosen. In tables 5.3 and 5.4, *Precision* and *Recall* values are also given, where *Precision* is the percentage of *Correct* with respect to *Detected*, and *Recall* is the percentage of *Correct* with respect to *Groundtruth*.

In addition to these metrics, valuable information can be obtained by determining *why* and *where* pedestrians were not detected. A summary of *why* pedestrians were not detected in each of the datasets is presented in tables 5.5 and 5.6. The reasons why a pedestrian was not detected are divided into 3 classes; *Occlusion*, represents a missed pedestrian due to high occlusion (where only a pedestrians head or less is visible in the scene); *Underseg.*, represents a missed pedestrian due to it becoming clustered with a second pedestrian into a single region (i.e. due to the under-segmentation of regions); and *Other* represents any other reasons, which are usually due to (1) a lack of accurate disparity or foreground edge information resulting in the region being removed as background, or (2) a pedestrian appearing at the edge of the scene of one image and not appearing in the opposite image. In tables 5.5 and 5.6 two values, A_{B}, are provided where A represents the total number of pedestrians missed due to the given reason and $_{B}$ represents the percentage of A with respect to the total number of pedestrians missed for that sequence. For example, in table 5.6 row 7 column 3, 77 pedestrians were missed because of high occlusion, which equates to 43.75% of the 176 total number of missed pedestrians in the *Corridor* sequence.

Finally, an overview of *where* the proposed technique failed to correctly detect pedestrians is presented figures 5.9(a),(c) and (d) for the *Overhead*, *Corridor* and *Grafton* sequences respectively. In each of these figures the red dots represent the centroids of the bounding boxes of all the groundtruthed people that did not have a match in the data. Overlaying these dots onto a background image for each sequence, see figures 5.9(b),(d) and (f), can be a visual cue to "problem" areas within the image sequences. The results of each test dataset (and sequence) will now be discussed separately.

Synthetic Dataset The results from the synthetic dataset are provided in table 5.3. In addition, they are also presented visually in figure 5.10, where; (Row1) illustrates the bounding boxes of

Sequence	*Groundtruth*	*Detected*	*Correct*	*Precision*	*Recall*
Synthetic Total	97	95	93	97.89	95.88

Table 5.3: Synthetic results overview.

Sequence	*Groundtruth*	*Detected*	*Correct*	*Precision*	*Recall*
Grafton 1	666	617	565	91.57	84.83
Grafton 2	754	720	664	92.22	88.06
Grafton 3	555	510	440	86.27	79.28
Grafton Total	1975	1847	1669	90.36	84.51
Overhead Total	657	714	615	86.13	93.61
Corridor Total	1027	965	851	88.19	82.86
2D Total	3659	3526	3135	88.91	85.68

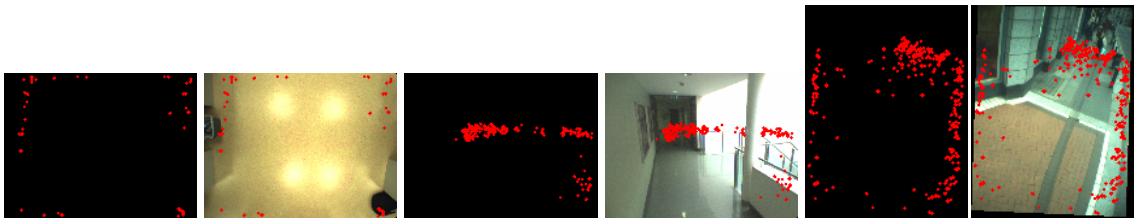
Table 5.4: 2D experimental results overview.

Sequence	*Total Missed*	*Occlusion*	*Underseg.*	*Other*
Synthetic	4	1 25	2 50	1 25

Table 5.5: Synthetic dataset missed pedestrians overview.

Sequence	*Total Missed*	*Occlusion*	*Underseg.*	*Other*
Grafton 1	101	23 22.77	9 8.91	69 68.32
Grafton 2	90	16 17.78	14 15.56	60 66.67
Grafton 3	115	8 6.96	5 4.35	102 88.7
Grafton Total	306	47 15.36	28 9.15	231 75.49
Overhead Total	42	0 0	2 4.76	40 95.24
Corridor Total	176	77 43.75	9 5.11	90 51.14
2D Total	524	124 23.66	39 7.44	361 68.89

Table 5.6: 2D missed pedestrians overview. Two figures, A $_{B}$, are provided for each of the columns 3-5. A represents the total number of pedestrians missed due to either Occlusion, Under-segmentation, or any other reason (usually lack of accurate disparity or foreground edge information). $_{B}$ represents the percentage of A with respect to the total number of pedestrians missed for that sequence. For example, for row 7 column 3, 77 pedestrians were missed because of high occlusion, which equates to 43.75% of the 176 total number of missed pedestrians in the *Corridor* sequence.



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

Figure 5.9: 2D Missed groundtruth persons; (a) and (b) *Overhead* sequence; (c) and (d) *Corridor* sequence; (e) and (f) *Grafton* sequences.

the pedestrians overlaid onto the stereo camera rig's right input image; (Row2) illustrates the final regions; and (Row3) illustrates the final pedestrian regions from a plan-view orientation, where the white lines indicate the bounds of the scene (that are defined with respect to the visible groundplane within the scene), the blue and yellow lines at the bottom centre of the image indicates the cameras position and orientation, and the position of detected pedestrians in that frame are illustrated by a circle. In each of these images, each pedestrian is allocated a specific colour whereby their bounding box, final region and plan-view orientation circle are all depicted using the same colour. It should also be noted that two separate pedestrians may have similar colours, however in order to ease visualisation the colours have been selected such that two pedestrians of the same colour do not appear beside each other.

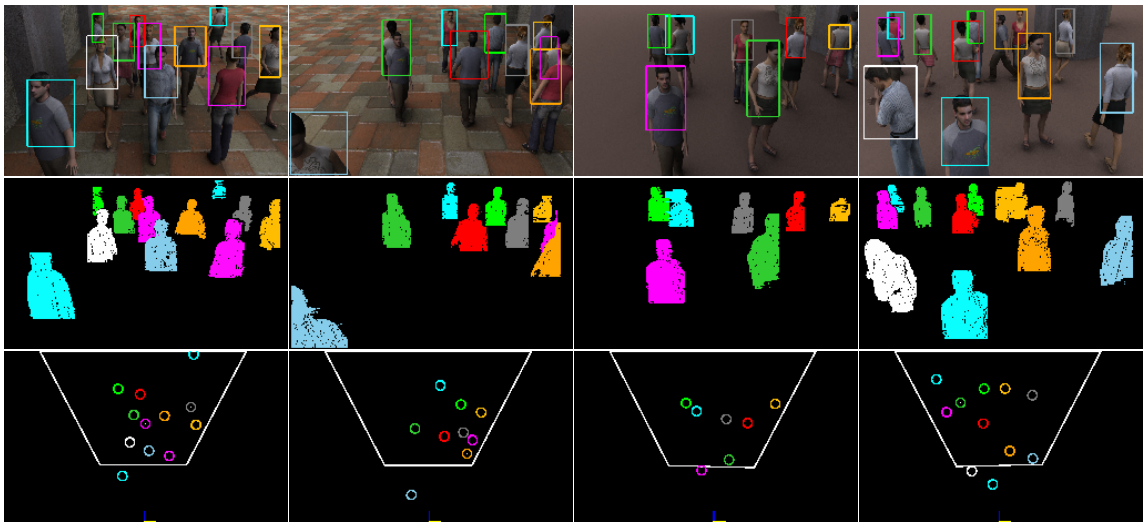
From tables 5.3 and 5.5 it can be seen that only 4 pedestrians were missed from the synthetic dataset; 2 in $F2$, one of which is missed as it is clustered with a second pedestrian, the other is missed due to high occlusion (this second person is positioned to the top right, close to the wall); 1 is missed in $F3$ (again positioned to the top right, close to the wall), but unlike before more of the pedestrian's body is present and so is not occluded, however, the disparity information for the majority of this person was not available as very little of the pedestrian can be seen in the opposing image; and finally a pedestrian in $F3$ is missed as it is clustered with a second pedestrian. Using the proposed evaluation metric it can be seen that when two or more pedestrians become clustered together into the same region, then this technique results in 100% precision and 50% recall for that region. It can be argued that this should be 0% precision and 0% recall as the region does not contain a single pedestrian and therefore is incorrect, however, this evaluation is deemed to be overly harsh and the 50% recall gives a more balanced perspective as the resultant region does indeed contain a detected pedestrian. However, analysis of the results from all the datasets (presented in section 5.5) reveal that only 0.92% of the final detected regions consist of two pedestrians and none consist of three or more. Therefore, on average, removing a single

percentage point off the overall precision and recall values will more than compensate for the effects of this choice. In addition, the number of under-segmented pedestrians in each separate dataset is provided in tables 5.5, 5.6 and 5.9, allowing the reader to recalculate the precision and recall values for any dataset via the preferred metric.

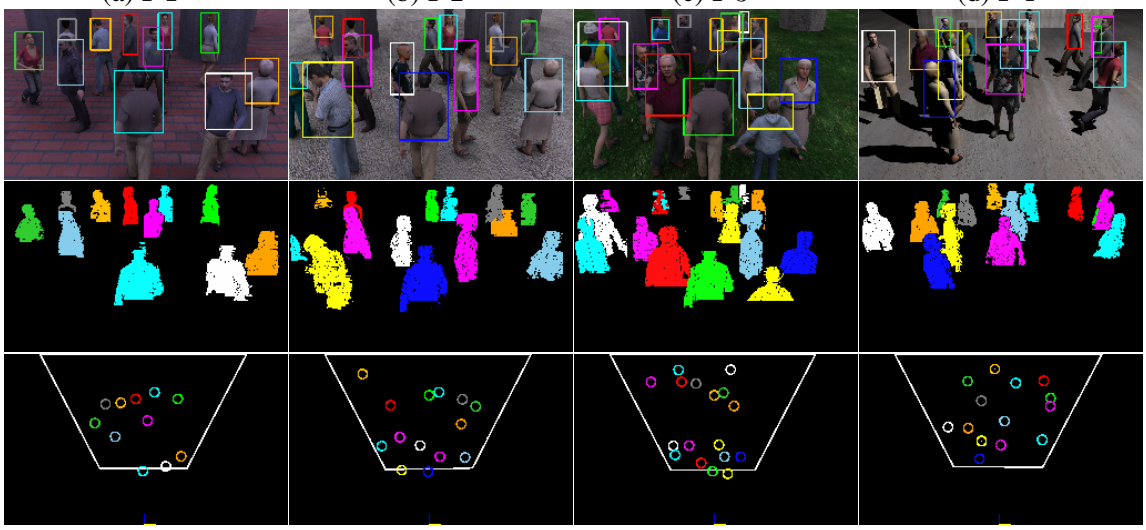
Finally, the precision value for the synthetic dataset is 97.89%, which resulted from two over-segmented regions, see $F7$ and $F8$. These over-segmented regions are generally due to either specific poses that result in a pedestrian exceeding the bounds of the pedestrian model or, in this case, due to incorrect disparity estimation. In general, the likelihood of the latter occurring increases with the distance of the pedestrian from the camera, as discussed in section 5.3.3. This type of artifact is difficult to remove using the proposed technique in single frame pedestrian detection, however the likelihood of them being detected as foreground pedestrians can be reduced using temporal information (as is seen in the following chapter).

Real-World Datasets The results from the real-world sequences are presented in tables 5.4 and 5.6, and some illustrative results from the 5 test sequences are presented in figures 5.11, 5.12, 5.13, 5.14 and 5.15. From tables 5.4 and 5.6 it can be seen that the *Overhead* sequence has the highest recall of 93.61%, this is to be expected as the pedestrians are closer to the camera and there is no full or partial occlusions of pedestrians by other objects (as expected for this sequence the number of pedestrians lost to occlusion is 0%). However, this sequence also has the lowest precision value of 86.13%, in addition it has the highest number of missed pedestrians due to "other reasons". Both of these two values are linked and can be explained with the use of figures 5.9(a) and (b).

Analysis of the pedestrians that are missed reveals that they are *all* positioned around the boundaries of the scene, at the points where people enter and exit the scene. It is not surprising that pedestrians are missed in these areas for two reasons; (1) the disparity is less likely to be well formed around the edges of the image, as discussed in section 4.4.3.2 and; (2) when a person enters the scene, the first portion of their body that enters the scene is likely to be a hand or their lower torso, followed shortly by their head and shoulders. Therefore, when entering and exiting the scene the regions observed by the camera are lower to the ground and clustering of regions with the golden ratio can result in a lower absolute value of δ, which controls the maximum clustering distance in the pedestrian detection module. This means that a large enough value of δ to cover the entire span of a human body may not be created until the shoulders and head enter the scene. This leads to increased over-segmentation or the removal of the smaller regions via the pedestrian



(a) $F1$ (b) $F2$ (c) $F3$ (d) $F4$



(e) $F5$ (f) $F6$ (g) $F7$ (h) $F8$

Figure 5.10: Synthetic data pedestrian detection results. For each set; (Row 1) Pedestrian bounding box; (Row 2) Pedestrian regions; (Row 3) Pedestrian centres in plan-view orientation.

detection module's post-processing steps. An illustrative example of this can be seen in figure 5.11(c), notice how the person at the bottom right of the image is split into two regions. However, two frames later in figure 5.11(d) the persons head has entered the scene, thereby increasing the absolute value of δ and allowing one correct region to be created. In can be seen in figures 5.9(a) and (b) that ignoring the pedestrians missed around the edges would lead to an almost 100% detection rate for the proposed technique.

From table 5.6 it can be clearly seen that the number of pedestrians missed due to occlusion is significantly higher in the *Corridor* sequence than in any other example scenario in both the total number (73) and the overall percentage (43.75%). This is simply explained by the camera height being the lowest of all 3 scenarios, therefore increasing the likelihood of occlusion, for example in

figure 5.12(f) the person on the left is occluding two other people from the camera's field of view. In addition, the pillar on the right hand side of the scene also occludes anyone who walks behind it. However, occlusion is not the biggest factor for missed pedestrians in the sequence as 51.14% of all missed pedestrians is caused by "other reasons", the cause of which is now explained.

Whilst the person interactions in this *Corridor* sequence are not particularly challenging, the sequence is interesting in terms of the distance at which these occur (which is greater than that of any of the other sequences) and the lighting conditions. The right hand of the scene is very bright whilst the left is much darker. Therefore, if people wearing brightly coloured clothes are on the right there is little texture information and vice versa on the left. This lack of texture has a degrading effect on the quality of the disparity and the pedestrian detection post-processing, so 3D regions in these areas are clustered less effectively and are more likely to be removed as background in post-processing. From figures 5.9(c) and (d) it can be seen that most of the missed pedestrians tend to congregate around the 8 metre mark, the entry point of the scene (at the bottom right) and the brightest or darkest regions of the image. In addition, people on the stairs tend to be missed as the regions become closer to the groundplane and therefore, as in the *Overhead* sequences, has a lower absolute value for δ. However, the precision (88.19%) and recall (82.86%) values are still very respectable for this difficult scenario, and it should be noted that many of these issues with distance and texture are not unique to the proposed technique as other techniques that rely on disparity, foreground segmentation or image gradients would be similarly effected.

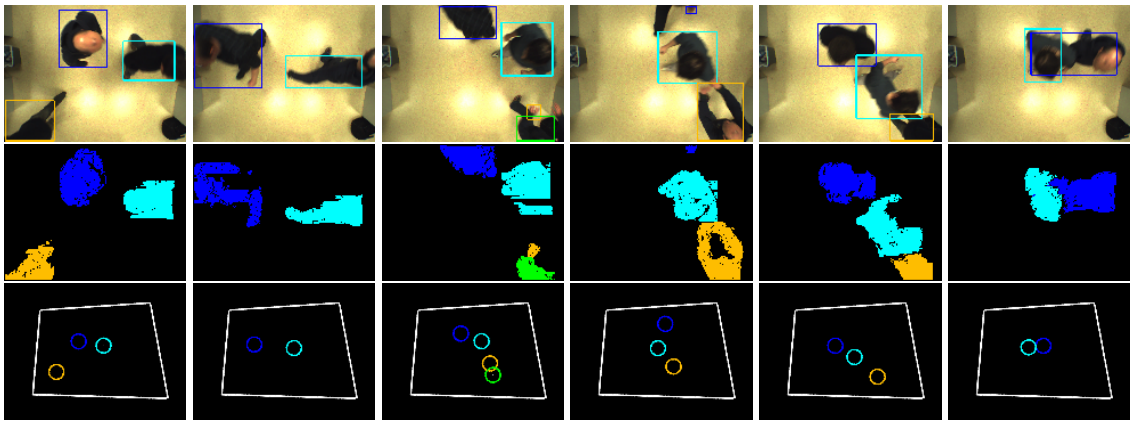
Finally, the *Grafton* sequences depicted in figures 5.13, 5.14 and 5.15 illustrate the robustness of the detection and tracking techniques when subjected to unconstrained crowded conditions. They all depict multiple pedestrians travelling in various directions being detected robustly. For *Grafton* sequences 1 and 2, the pedestrian flow can be significantly higher than any other sequence employed, with up to 13 people being correctly detected in a single image, see figure 5.14(b). For the three *Grafton* sequences, the precision values were consistently better than the other two sequences (averaging 90.36%), while the recall values averaged 84.51% therefore lying in between the recall values for the other two sequences. Figures 5.9(e) and (f) back up the observations made for other sequences, whereby the vast majority of pedestrians missed tend to congregate either around the 8 metre mark or at the scene boundaries.

In addition to being able to detect moving pedestrians, *Grafton* sequence 2 illustrates how a pedestrian whom is standing still for a large period of time can be robustly detected due to the slow update parameter of the background gradient model. For many techniques this slow update would

cause significant problems when faced with rapidly changing lighting conditions. However, as the proposed technique applies the background model in such a manner that only guides the subtraction of foreground objects, the resultant technique becomes very robust to background or global illumination changes. The robustness of this technique for background subtraction is illustrated in the results from *Grafton* sequence 3, which incorporates rapidly changing lighting conditions – see figure 5.15. In this challenging sequence, due to the layered background modelling framework and the length of time required for changes to the background to remain in the cache is longer than the total number of frames in the test sequence, the background model remains effectively unadaptive to any sudden background or global illumination changes – see section 4.3.6.1. However, even with an effectively static background model, the algorithm performed extremely robustly maintaining precision (86.27%) to a level that is comparable to other sequences whereby the lighting conditions were constant.

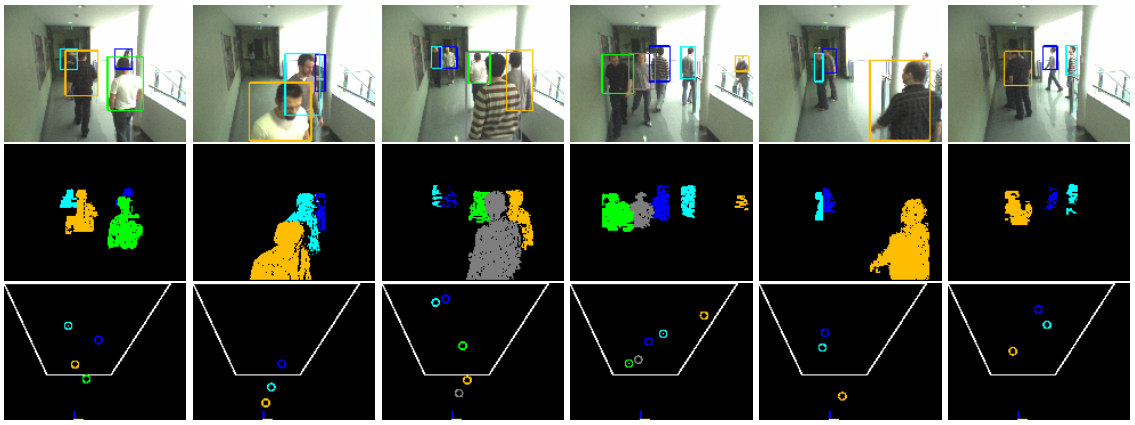
Although shadows do not cause a problem to system precision, the recall is affected by the strong shadows caused by buildings. For *Grafton* sequence 3 the recall value (79.28%) was slightly lower than all other sequences. This drop in recall is due to a lack of texture in the input image. This effect can also be seen in table 5.6, where the number of pedestrians missed due to "other reasons" rises from 68.32% and 66.67% in Grafton sequences 1 and 2 respectively to 88.7% in Grafton sequences 3. The reason for these extra missed pedestrians is illustrated in figures 5.16(a) and (b) where the missed pedestrians are outlined in red. Due to the strength of the cast shadow, little foreground gradient information can be extracted from the input image and therefore the pedestrians in shadows are removed as background. This problem however does not only occur in shadow, see figure 5.16(c) where the person in red is again missed due to lack of texture between foreground and background objects.

The precision from these *Grafton* sequences is very high, averaging at 90.36%, therefore only 9.64% of regions detected have no groundtruth match. Further evaluation indicates that the majority of these "extra regions" are be caused by the over-segmentation of pedestrian regions, see figures 5.16(d) and (e) where the red circle in the middle row indicates which pedestrian has been over-segmented. In addition, a very small number of background objects are incorporated into this percentage, for example in figures 5.16(c) where the false-positive is due to a reflection of a pedestrian in a window. Finally, the robustness of the system to under-segmentation can be seen in the fact that the 1975 pedestrians detected in the *Grafton* sequences only 28 of these (or 1.26%) of the regions were under-segmented, an example of under-segmentation can be seen in figure



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

Figure 5.11: *Overhead* sequence, frame numbers; (a) 69; (b) 102; (c) 184; (d) 186; (e) 222; (f) 348; (Row 1)-(Row 3) as in figure 5.10.



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

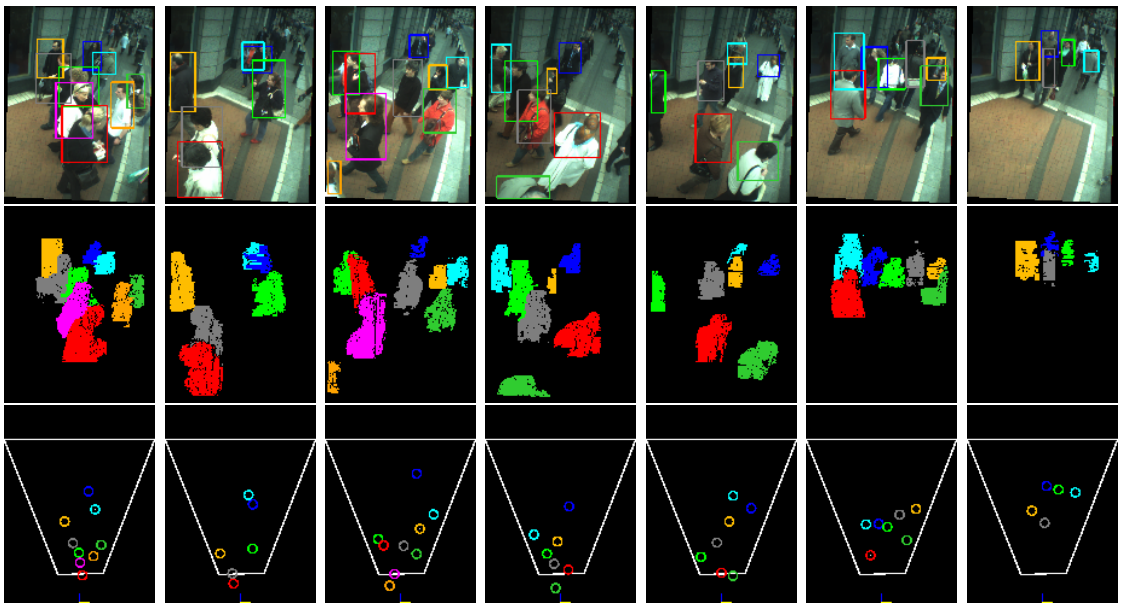
Figure 5.12: *Corridor* sequence, frame numbers; (a) 28; (b) 238; (c) 286; (d) 299; (e) 371; (f) 390; (Row 1)-(Row 3) as in figure 5.10.

5.16(g).

5.4.2 3D Evaluation

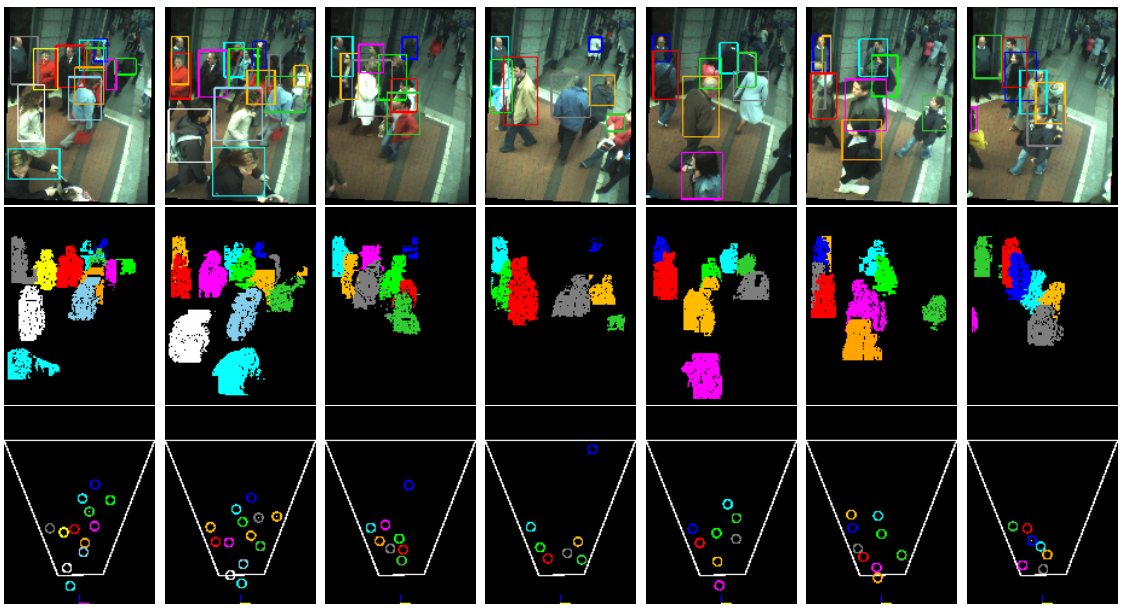
We acknowledge that there are some issues with the 2D groundtruth evaluation process. They include;

- The technique only groundtruths pedestrians that are less than 8 metres from the camera, but when is a person exactly 8 metres from the camera? It is difficult to determine in a 2D image if a person is 7.5, 8 or 8.5 metres from the camera. In the test sequences an imaginary bounding line is drawn across the groundplane based on measurements taken from the scene. Anyone that crosses this line is within the required distance. This is not ideal however, as occlusions due to crowds can lead to empirical estimation of how far a person is from the



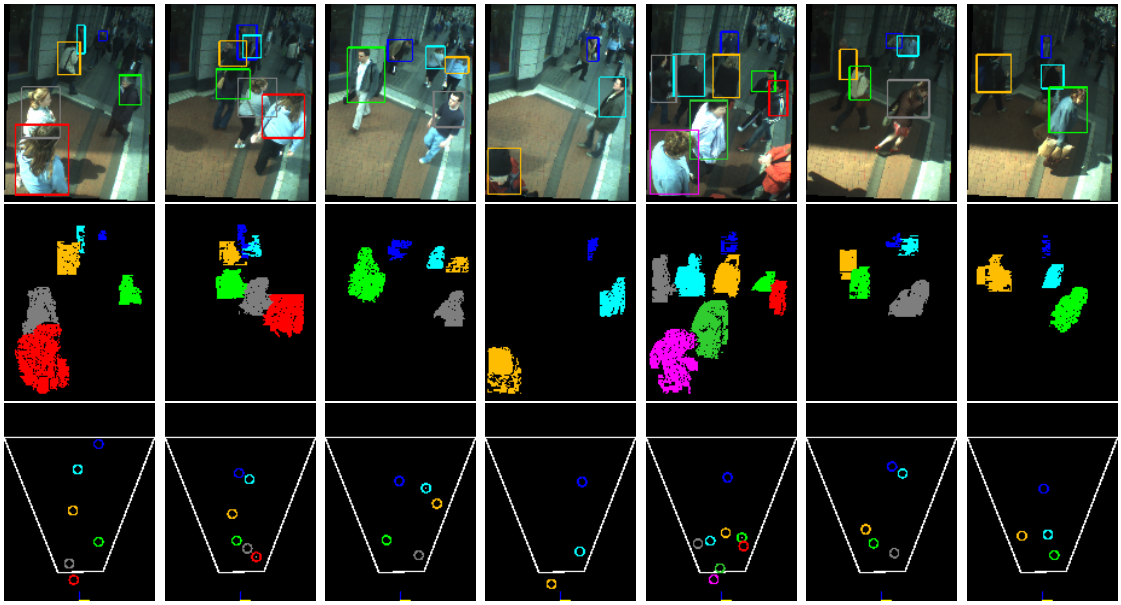
(a) (b) (c) (d) (e) (f) (g)

Figure 5.13: *Grafton* sequence 1, frame numbers; (a) 12; (b) 30; (c) 37; (d) 41; (e) 50; (f) 87; (g) 96; (Row 1)-(Row 3) as in figure 5.10.



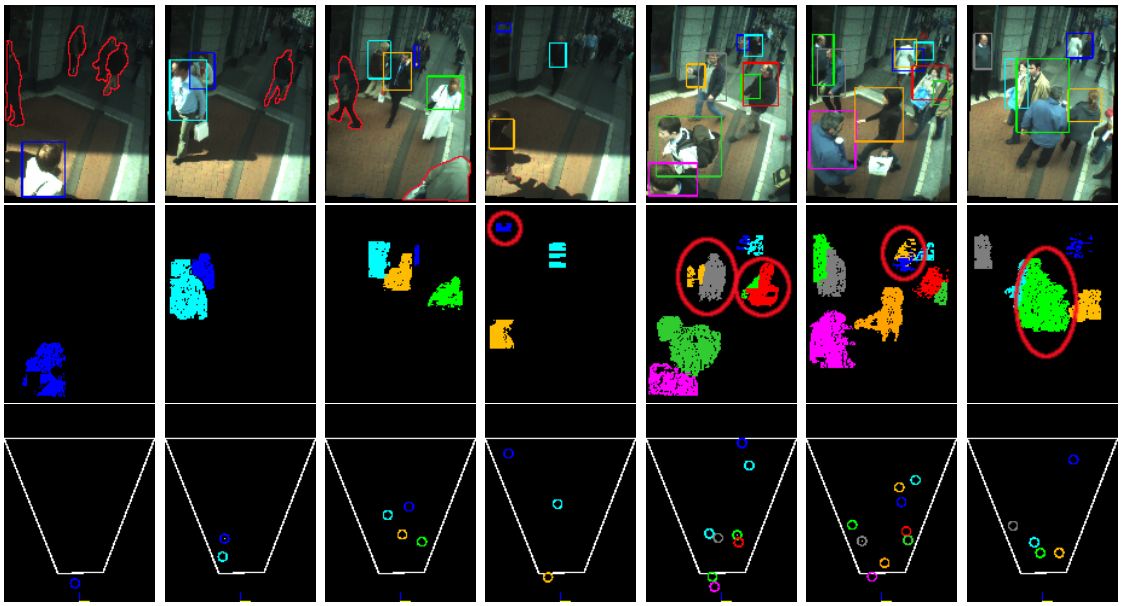
(a) (b) (c) (d) (e) (f) (g)

Figure 5.14: *Grafton* sequence 2, frame numbers; (a) 24; (b) 26; (c) 40; (d) 56; (e) 69; (f) 85; (g) 88; (Row 1)-(Row 3) as in figure 5.10.



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

Figure 5.15: *Grafton* sequence 3 using a single static background gradient model. Frame numbers; (a) 26; (b) 32; (c) 45; (d) 51; (e) 64; (f) 85; (g) 104; (Row 1)-(Row 3) as in figure 5.10.



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

Figure 5.16: *Grafton* sequences issues; (a)-(c) No foreground edges; (d) Specular reflections; (e)-(f) Over-segmentation; (g) Under-segmentation; (Row 1)-(Row 3) as in figure 5.10.

camera.

- How accurate is the system for a maximum distance of 5, 6, 7 or 8 metres? Does the system's performance degrade gradually or is there a threshold distance after which there a large drop off in performance?
- Is a 50% overlap between test and groundtruth data bounding boxes reliable enough?
- How accurate are the 3D statistics obtained from each pedestrian such as height and 3D position?
- How accurate is the observation that missed pedestrians tend to occur towards the 8 metre mark and at the edges of the input image?
- How does the performance of the system perform with respect to varying pedestrian numbers?

5.4.2.1 Vicon Setup

To help answer some of these questions, the system has also been evaluated against groundtruth data captured using a 3D Vicon infrared motion analysis system [206] developed by Oxford Metrics Group Ltd. The Vicon system is an automated motion capture system that tracks the position of infra-red reflective markers in 3D space. The Vicon system has been extensively used in sports science, animation and technology applications and is in use in a wide variety of leading visual effects houses and games companies, including Nintendo, Industrial Light and Magic, and Sony [206]. The Vicon system offers a high degree of accuracy in 3D, up to 1mm in 6m space. A Vicon infra-red camera can be seen in the right of figure 5.17(a). For these experiments 12 such cameras were employed.

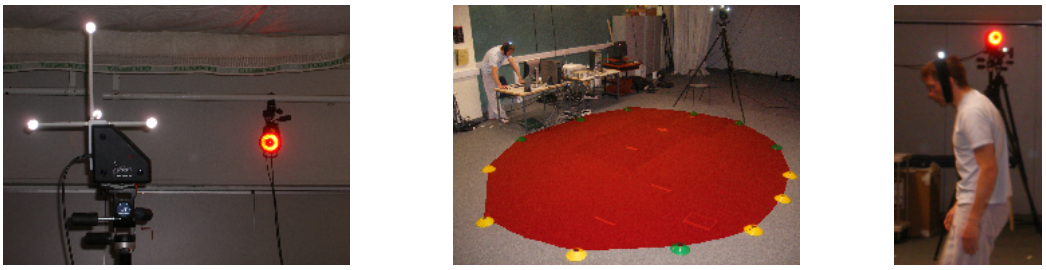
We have carried out a number of experiments using the Vicon system to obtain the groundtruth data. In these experiments, the Digiclops camera was fitted with a local co-ordinate system, see figure 5.17(a), consisting of infra-red reflective markers that can be accurately located in 3D via the Vicon system. These co-ordinate system markers indicate the precise position and orientation of the Digiclops camera with respect to a global Vicon co-ordinate system. In addition, the 3D position of the groundplane was obtained via markers in the global Vicon co-ordinate system. During the experiments, each person was accurately and reliably tracked in 3D via the Vicon system using infra-red reflective markers situated on the top of each persons' head, see figure

5.17(c). Using the Vicon co-ordinate system the height above the groundplane of each of these points was also obtained.

In our experiments, the position of a detected pedestrian within the proposed system was defined to be the centre of mass of a detected pedestrian's *head* region orthographically projected onto the groundplane. The head region feature point was chosen as; (1) it is generally more robust and accurate than other features, such as the centre of mass of a full body region, during occlusion; and (2) the Vicon coordinate is also located at the centre of the head. This feature point is obtained by using the biometric model of section 5.3.1 and by calculating the centre of mass of all 3D points belonging to a detected pedestrian that lie between the bounds of $|af|$ (which is the distance from the top of the head to the base of the skull). The feature point was then orthographically projected onto the groundplane as; (1) this is the positional feature for pedestrians adopted by the tracking module described in chapter 6, as tracking pedestrian positions with respect to the groundplane is favourable when compared to that of tracking in 3D with respect to an arbitrary 3D world origin; and (2) vertical offsets in position features, such as the difference between the *centre of mass* of a detected pedestrian's head and a Vicon marker's on *top* of a pedestrian head, should not be incorporated into the position evaluation metric (a separate height metric evaluates these differences). Therefore, each Vicon groundtruth 3D head position is orthographically projected onto the groundtruth Vicon groundplane. Finally, via a rotation and transformation, each of the groundtruth pedestrian feature points was transformed from the Vicon global co-ordinate system to the Digiclops's local co-ordinate system.

Using this data, a comparison between groundtruth 3D data and detected pedestrians' 3D statistics is possible. Using this evaluation information *some* of the outstanding questions from the 2D evaluation process can be answered. Unfortunately due to limitations in the Vicon system's field of view not all of the required data to answer all the outstanding questions could be gathered. In reality, the Vicon system provided an elliptical area of reliable tracking that measured 5.5 metres in length and 3.15 metres in width with respect to the Digiclops's principal axis – see the area highlighted in red in figure 5.17(b). As pedestrians were therefore limited to a maximum distance of 5.5 metres from the camera some of the outstanding questions relating to the proposed technique's performance at its maximum distance of 8 metres could not be addressed. As part of future work it is intended to recalibrate the Vicon system in a larger room thereby incorporating a larger field of view to answer some of the remaining questions.

For this evaluation, five different test sequences were used; the first sequence, *Vicon* 1, con-



(a) (b) (c)

Figure 5.17: Vicon system setup; (a) Digiclops camera with co-ordinate system (left) and Vicon camera (right); (b) Vicon capture area highlighted in red; (c) Groundtruth reflective marker.

Sequence	*Num. Pedestrians*	*Num. Frames*	*Hz*	*Time (in Minutes)*
Vicon 1	198	198	≈ 5.4	0.86
Vicon 2	526	263	≈ 5.5	0.94
Vicon 4	1296	324	≈ 5.3	1.18
Vicon 8_A	2104	263	≈ 5.4	0.96
Vicon 8_B	2120	265	≈ 5.4	1.00
Total	6244	1313	≈ 5.3–5.5	4.94

Table 5.7: 3D dataset sequences.

sisted of 1 person; the second sequence, *Vicon* 2, consisted of 2 people; the third sequence, *Vicon* 4, consisted of 4 people; and the final two sequences, *Vicon* 8_A and *Vicon* 8_B, consisted of 8 people. An overview of these sequences is provided in table 5.7. Each sequence was set in an indoor setting with the camera positioned just above 2 metres from the ground. This setup is similar to that of the *Corridor* scene with regards to camera placement and orientation. The lighting conditions are stable, however, as with the *Overhead* scene, the groundplane is highly specular and brightly illuminated.

5.4.2.2 Evaluation

Precision and recall values were obtained for each sequence, see table 5.8, where a match between a detected pedestrian and a groundtruth pedestrian was determined via the 3D information and the biometric pedestrian model introduced in section 5.3.1. Using this biometric model a value of $|\text{lo}|$ (i.e. the width of the shoulders) was determined for each groundtruth person's position. If the proposed technique detects a person within a Euclidean distance of this value $|\text{lo}|$ to the groundtruth position then a match was made between the detected and groundtruth pedestrian, i.e. if a pedestrian is detected via the proposed technique *and* its reconstructed position is within shoulder distance of a groundtruth position, it is considered a good match.

In addition to the precision and recall values, an evaluation of *why* and *where* pedestrians were

Sequence	*Groundtruth*	*Detected*	*Correct*	*Precision*	*Recall*
Vicon 1	198	205	198	96.59	100
Vicon 2	526	533	523	98.12	99.43
Vicon 4	1296	1277	1248	97.73	96.3
Vicon 8_A	2104	1899	1874	98.68	89.07
Vicon 8_B	2120	1859	1843	99.14	86.93
3D Total	6244	5773	5686	98.49	91.06

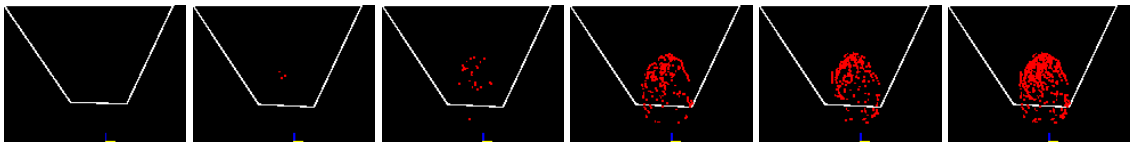
Table 5.8: Experimental results overview.

Sequence	*Total Missed*	*Occlusion*		*Underseg.*		*Other*	
Vicon 1	0	0	0	0	0	0	0
Vicon 2	3	3	100	0	0	0	0
Vicon 4	48	46	95.83	1	2.08	1	2.08
Vicon 8_A	230	197	85.65	11	4.78	22	9.57
Vicon 8_B	277	230	83.03	29	10.47	18	6.5
Total	558	476	85.3	41	7.35	41	7.35

Table 5.9: 3D missed pedestrians overview.

missed was also carried out, see table 5.9 and figure 5.18. However figure 5.18 is displayed in a plan-view orientation as it is a more informative presentation of the positions where pedestrians were missed. In addition, the accuracy of the position and 3D height statistics for the detected pedestrians were calculated and presented in tables 5.10 and 5.11 respectively. Finally, illustrative example results are presented in figures 5.19, 5.20 and 5.21, where; (Row1) illustrates the bounding boxes of the pedestrians overlaid onto the input images; (Row2) illustrates the final regions; (Row3) illustrates the final pedestrian regions from a plan-view orientation; and (Row4) illustrates the *groundtruth* pedestrian regions from a plan-view orientation where a green circle indicates a match was made, and a red circle indicates no match was possible.

It can be seen from table 5.8 that the *minimum* precision of any of the Vicon sequences was 96.59%, which is still 4.37% better than the precision of any of the previously examined sequences. In addition, the recall from these sequences is very high with an average recall from the 5 sequences of 91.06%. However, if the 476 pedestrians that were missed due to occlusion were removed, see table 5.9, then the recall would be on average 98.58%. This result reinforces the belief that many of the missed pedestrians in the previous 2D sequences occur at the edges of the scene and towards the 8 metre mark, which is the perceived limit in distance of the proposed system. In these experiments, due to the limited field of view of the Vicon system, pedestrians were kept to a closer distance to the camera (the maximum distance recorded of a pedestrian to the camera was 5.47 metres – see table 5.10) and within the field of view of the stereo camera. This



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

Figure 5.18: 3D missed groundtruth persons; (a) *Vicon* 1; (b) *Vicon* 2; (c) *Vicon* 4; (d) *Vicon* 8_A; (e) *Vicon* 8_B; (f) All *Vicon* sequences.

Sequence	*Min. Distance*	*Max. Distance*	*Av. Distance*	*Av. Error*	*Av. % Error*
Vicon 1	227.84	514.18	382.42	10.76	2.81
Vicon 2	246.34	500.49	357.44	11.75	3.29
Vicon 4	229.75	538.48	357.43	8.11	2.27
Vicon 8_A	217.69	539.78	351.66	9.81	2.79
Vicon 8_B	220.16	547.47	362.19	7.14	1.97
Total	217.69	547.47	357.94	8.78	2.45

Table 5.10: Distance experimental results overview (in cm).

therefore eliminated many of the missed detections recorded in the previous experiments.

For the *Vicon* 1 sequence 100% recall was achieved, however, as the number of people were increased the number of missed pedestrians also increased, see figures 5.18 where the increase in missed pedestrians between (a)-(d) can be clearly seen. However, unlike previous examples the missed pedestrians are more evenly spread throughout the oval region in which pedestrians were present, see figure 5.18(f). This is explained using table 5.9 where it can be seen that 85.3% of all missed pedestrians are due to occlusions, and as an occlusion can happen anywhere in the scene the distribution of points becomes more even (it should be noted however that most of the points closer to the camera are missed pedestrians due to either under-segmentation or "other reasons"). Examples of occlusions can be seen in figures 5.19(d), 5.20(e) and 5.21(c). In addition to increased pedestrians missed due to occlusions, an increase is recorded in table 5.9 for both other cases of missed pedestrians. This is due to a reduction in the average space within the region for each person as the number of pedestrians increases. This therefore forces pedestrians into closer proximity and to be pushed farther towards the edges of the Digiclops camera's field of view. An example of under-segmentation in the *Vicon* 8_B sequence is given in figure 5.21(f). Although the recall decreased when more people were added to the scene, precision remained constant and actually increased slightly as more people were added. This effect may be due to occlusions of pedestrians at farther distances to the camera, therefore decreasing the average distance of detected pedestrians could result in less over-segmentation on distant pedestrians.

Finally, the 3D statistics of each detected pedestrian were compared to their corresponding

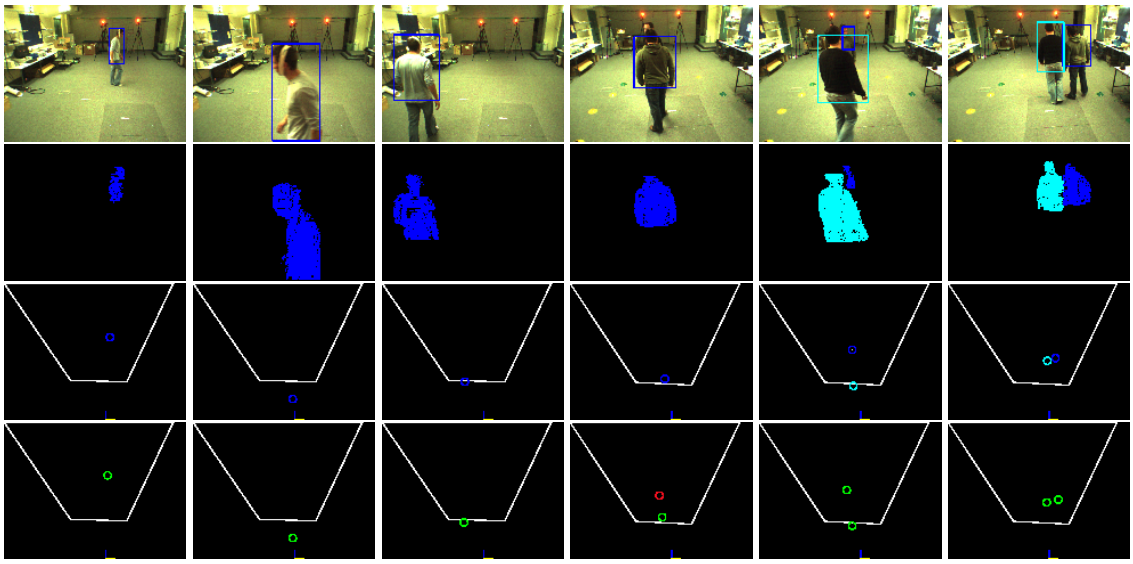
Sequence	*Min. Height*	*Max. Height*	*Av. Height*	*Av. Error*	*Av. % Error*
Vicon 1	178.09	183.05	180.92	7.27	4.02
Vicon 2	170.98	188.03	181.56	8.4	4.63
Vicon 4	166.15	191.86	181.69	9.52	5.24
Vicon 8_A	154.37	188.95	177.61	9.97	5.61
Vicon 8_B	148.52	218.83	177.84	10.22	5.75
Total	148.52	218.83	179.06	9.71	5.42

Table 5.11: Height experimental results overview (in cm).

groundtruths, the results of which are presented in tables 5.10 and 5.11 (it should be noted that the maximum and minimum heights of *Vicon* 8_A in table 5.11 are significantly different to those in *Vicon* 8_A as one person crouched and then jumped within the sequence). It can be seen that the average error in positioning the pedestrians (with respect to the Digiclops system) was only 8.78 cm and the average error in height was 9.71 cm. This equates to a 2.45% and 5.42% error in the average positioning and height respectively. The greater error in height was expected due to the pedestrian detection post-processing technique clipping the bounding box of a region, which affects height more dramatically than the centre of mass of a pedestrian's body. It should be noted that these error percentages incorporate errors in the calibration of the groundplane and both camera systems, the 3D reconstruction from the Digiclops camera, and the proposed pedestrian detection technique. In addition, the height statistics include a 1.5cm offset from the top of a pedestrian's head to the centre of the reflective Vicon marker positioned *on top* of their head. From these results it is shown that these values are relatively consistent and highly accurate with respect to the number of pedestrians and occlusions within the scene.

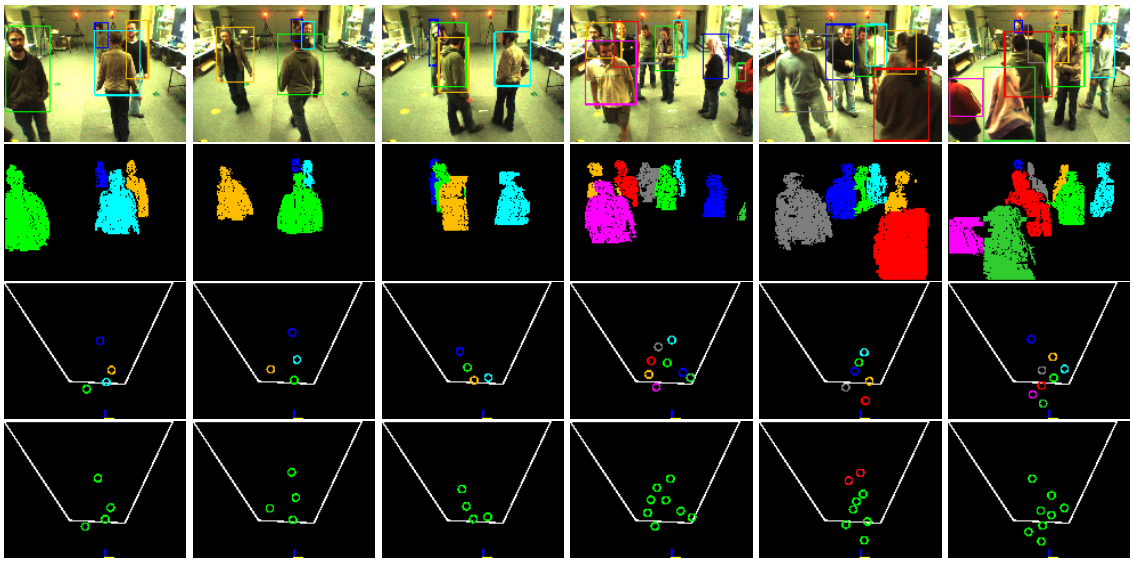
5.5 Summary

In this chapter a pedestrian detection technique was presented that clusters 3D points, obtained from a post-processed disparity map, into coherent pedestrian regions via an iterative region growing framework. This technique was shown to be robust to a variety of camera heights, rotations and orientations and also to varying pedestrian flow and illumination conditions. The technique was evaluated using 10,000 groundtruth pedestrian regions, see table 5.12, using two differing methodologies, against a number of test sequences from a variety of scenes. Results from this evaluation illustrate the technique's robustness to both over-segmentation (the average precision was 94.89%, see table 5.12) and under-segmentation (only 82 or 0.92% of detected pedestrians



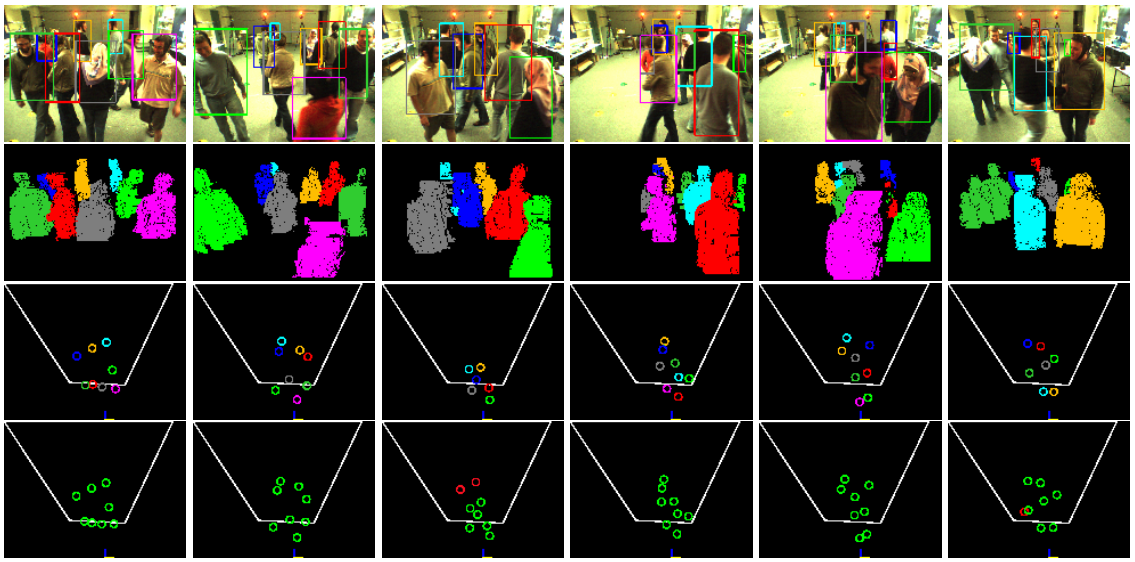
(a) (b) (c) (d) (e) (f)

Figure 5.19: *Vicon* 1 (a-c) and *Vicon* 2 (d-f) sequences, frame numbers; (a) 90; (b) 115; (c) 125; (d) 71; (e) 96; (f) 213. For each set; (Row 1) Pedestrian bounding box; (Row 2) Pedestrian regions; (Row 3) Detected pedestrian centres in plan-view orientation; ; (Row 4) Groundtruth pedestrian centres in plan-view orientation, where a green circle indicates a match was made and a red circle indicates no match was possible.



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

Figure 5.20: *Vicon* 4 (a-c) and *Vicon* 8_A (d-f) sequences, frame numbers; (a) 35; (b) 71; (c) 278; (d) 34; (e) 65; (f) 103; (Row 1)-(Row 4) as in figure 5.19.



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

Figure 5.21: *Vicon* 8_A (a-c) and *Vicon* 8_B (d-f) sequences, frame numbers; (a) 120; (b) 157; (c) 186; (d) 151; (e) 175; (f) 210; (Row 1)-(Row 4) as in figure 5.19.

were under-segmented, see table 5.13). In addition, the technique obtained an average recall value of 89.14%. This is a very high recall value for extremely challenging data where not all the pedestrians are visible in each image. This recall value rises to 94.84% if the 601 highly occluded pedestrians are removed from the groundtruth numbers.

Robust pedestrian detection in individual frames is a basic requirement of many pedestrian tracking techniques, whether they be *continuous detect-and-track* or *single detect-and-track* techniques (see section 1.3.2). In the proposed pedestrian tracking approach a *continuous detect-and-track* technique is chosen. This choice of tracking methodology is only possible for the challenging test sequences if the precision and recall values for the pedestrian detection technique are high enough. However, taking into consideration these evaluation metrics from a variety of scenarios, it is believed that robust tracking via this methodology is possible. This pedestrian tracking technique, presented in the following chapter, is the third contribution of this thesis and the final module in the overall system (see figure 3.17). Using this technique detected pedestrians are temporally tracked by representing previous tracks and current image pedestrians by a *Weighted Bipartite Graph*. A *Maximum Weighted Maximum Cardinality Matching* scheme is then employed, with additional kinematic constraints, to obtain the best match from previous tracks to currently detected pedestrians. A number of separate rollback loops are used to backtrack the pedestrian detection module to various states to further reduce over-/under-segmentation of detected pedestrians and increase tracking robustness. The final pedestrian detection and tracking

Sequence	*Groundtruth*	*Detected*	*Correct*	*Precision*	*Recall*
Synthetic Total	97	95	93	97.89	95.88
2D Total	3659	3526	3135	88.91	85.68
3D Total	6244	5773	5686	98.49	91.06
Total	10000	9394	8914	94.89	89.14

Table 5.12: Experimental results overview.

Sequence	*Total Missed*	*Occlusion*	*Underseg.*	*Other*
Synthetic Total	4	1 25	2 50	1 25
2D Total	524	124 23.66	39 7.44	361 68.89
3D Total	558	476 85.3	41 7.35	41 7.35
Total	1086	601 55.34	82 7.55	403 37.11

Table 5.13: Missed pedestrians overview.

technique are evaluated against the same challenging datasets that were employed in this chapter.

CHAPTER 6

Pedestrian Tracking

6.1 Introduction

The pedestrian detection technique described and evaluated in chapter 5 is designed to segment pedestrians within single frames. However, the goal of many computer vision applications, such as those used for security, obtaining pedestrian flow information and smart rooms, require information about not only where people are at a given instant in time, but also *what* these people are doing. The information obtained at a discrete time instant is unlikely to provide this or other reliable information about the activities of people within the scene. Clearly what is required in these cases is to robustly track pedestrians through time. This information could potentially be used in other layers of an application's framework to aid pedestrian activity classification.

As discussed in section 2.4 there are two broad tracking methodologies. The first is a *continuous detect-and-track* approach whereby pedestrian detection techniques are employed in each separate frame. Each of the detected pedestrians is then matched to previous tracks via some similarity measure. The second set of techniques can be thought of as *single detect-and-track* approaches, whereby once the detection is made an independent tracking scheme is employed. For these tracking based techniques the number of pedestrians in the scene and knowledge of their appearance is known *a priori*, this information is then used to segment the pedestrians in the current frame. In these types of techniques, unexplained foreground regions are usually considered to belong to a single *new* pedestrian.

In this chapter, the third module of the proposed system, and the third major contribution of this thesis, is presented. This module provides a *continuous detect-and-track* framework in which pedestrians are temporally tracked using both appearance and positional information. A

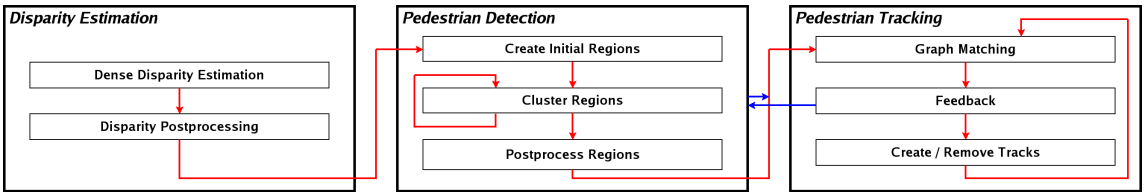


Figure 6.1: Pedestrian detection and tracking system overview.

high-level overview of this module can be seen in figure 6.1. The association between previous tracked pedestrians and pedestrians detected in the current frame, outlined in section 6.2.1, is made by representing the two sets of pedestrians as a *Weighted Bipartite Graph*, where the weights are determined via appearance models and other heuristics that ensure that more established tracks in the system are given priority over younger tracks that may be caused by noise. A *Maximum Weighted Maximum Cardinality Matching* scheme with additional kinematic constraints, presented in section 6.2.2, is then employed to obtain the best match from previous tracks to currently detected pedestrians. Finally, a number of separate rollback loops, described in section 6.2.3, are used to backtrack the pedestrian detection module to various states to further reduce over-/under-segmentation of detected pedestrians and increase tracking robustness.

In section 6.3 the proposed pedestrian tracking technique is evaluated against the same datasets that were employed in chapter 5. This evaluation is made to highlight the benefits of the rollback loops with respect to precision and recall values and to examine the robustness of the pedestrian tracking technique. Finally, section 6.4 provides a discussion on the proposed tracking approach, its evaluation and the pedestrian detection and tracking system as a whole.

6.2 Proposed Pedestrian Tracking Technique

The proposed pedestrian tracking technique is based on a *continuous detect-and-track* methodology. Pedestrians are detected in each frame individually via the pedestrian detection algorithm of chapter 5. These pedestrians are then associated to pedestrians detected in previous frames via appearance models and some predefined heuristics. This choice of tracking methodology was chosen over a *single detect-and-track* technique for a number of reasons, most of which stem from the fact that object segmentation has already been implemented. These include;

- less sophisticated and detailed appearance models are required. This is advantageous as the models can be made more robust to changes in pedestrian pose and scale;

- appearance model update procedures are less critical;
- the technique is less reliant on prediction techniques that are required to determine a previously tracked object's position in the current frame. This is advantageous in real-world scenarios as a pedestrian may travel in a non-linear and unpredictable fashion, resulting in difficulties in accurate tracking prediction;
- explicit occlusion analysis does not have to be estimated *a priori*;
- it simplifies the process of creating and eliminating tracks;
- dynamic selection of pedestrian and track features can be applied after segmentation to eliminate ambiguous matches;
- difficulties due to temporal drift are eliminated;

A disadvantage to this approach is that *continuous detect-and-track* techniques must have sufficiently high precision and recall values from the underlying segmentation technique to make robust tracking possible. However, this can also be an issue with *single detect-and-track* methodologies. When a previously un-tracked pedestrian enters the scene it must be correctly segmented, via some similar underlying pedestrian detection technique. This segmentation is necessary in order to obtain the required appearance models that are used for tracking in subsequent frames. This can be difficult in unconstrained environments when pedestrians enter and exit the scene in groups, possibly occluded. In addition, many *single detect-and-track* techniques require a person to remain un-occluded for enough frames for the model to stabilise and accurately describe the colour of pixels within the full area of pedestrian movement [139].

In the proposed approach, tracking is implemented using both positional and colour appearance information. This positional information is determined from a 3D pedestrian feature point that is then orthographically projected onto the groundplane, thereby eliminating one degree of freedom. As discussed in section 3.5, tracking pedestrian positions with respect to the groundplane is favourable when compared to that of tracking in 3D with respect to an arbitrary 3D world origin. The reason for this is that it can significantly simplify the prediction of a pedestrian's position in the subsequent frames, for example it is inherently assumed that when a person moves within a scene, they move across a groundplane. Using 3D co-ordinates, however, a pedestrian model is free to move with an extra degree of freedom, allowing hypothesised positions of pedestrians to be illogically above or below a groundplane in subsequent frames. In addition, there is

no critical loss of information using this transformation, as the mapping between 3D to 2D can be a Euclidean transformation, which would allow Euclidean distances between tracks and pedestrians to be calculated. As described in section 5.4.2.1, the 3D pedestrian feature point that is orthographically projected onto the groundplane is chosen to be the centre of mass of a detected pedestrian's *head* region which is considered to be more robust and accurate than various other features during occlusions.

Using this positional information, plus colour and height appearance models, associations between currently detected pedestrians and previous tracks are made by representing the two sets as a *Weighted Bipartite Graph* (see section 6.2.1). The best match from previous tracks to currently detected pedestrians is made within this graph using a *Maximum Weighted Maximum Cardinality Matching* scheme (see section 6.2.2), which also employs additional kinematic constraints. In addition, in section 6.2.3, a number of separate rollback loops are applied to backtrack the pedestrian detection module to various states to further reduce over-/under-segmentation of detected pedestrians and increase tracking robustness. Finally, as described in section 6.2.4, tracks are post-processed with a view to increasing track stability with respect to pedestrian over-segmentation and occlusion problems. Each of these separate steps is now described in detail.

6.2.1 Weighted Bipartite Graph

Let $p_1, p_2, \ldots p_N$ represent the N pedestrians that have been detected by the pedestrian detection module in frame i, and let $t_1, t_2 \ldots t_M$ represent the M pedestrians that have been temporally tracked up to frame $i-1$. If $M = 0$ and $N > 0$, then each p_x is assigned a new track t_x, where $x = 1 \ldots N$. For all frames where $M > 0$, it is required to update the M tracks to incorporate pedestrian data from frame i. This is achieved by matching the N pedestrians in frame i to the M tracks from frame $i-1$. However, it may not be possible to match all pedestrians to tracks, or vice versa. In addition, as outlined in section 6.2.1.1, a given pedestrian may be more or less likely to be a continuation of a certain track.

This situation can be represented by a *weighted bipartite graph*, $G = (V, E)$ [212]. A graph is bipartite if there exists a partition of the vertex set $V = V_1 \cup V_2$ so that both V_1 and V_2 are independent sets, and an edge, $e_{v_1 v_2} \in E$, can only link $v_1 \in V_1$ to $v_2 \in V_2$. In this scenario, V_1 represents the N pedestrians detected in the current frame i, and V_2 the M temporally tracked pedestrians in frame $i-1$. e_{xy} denotes a match between a pedestrian, p_x, and a track, t_y, where $x = 1 \ldots N$ and $y = 1 \ldots M$. To match pedestrians to tracks, a subset of edges, $\hat{E} \subset E$, is

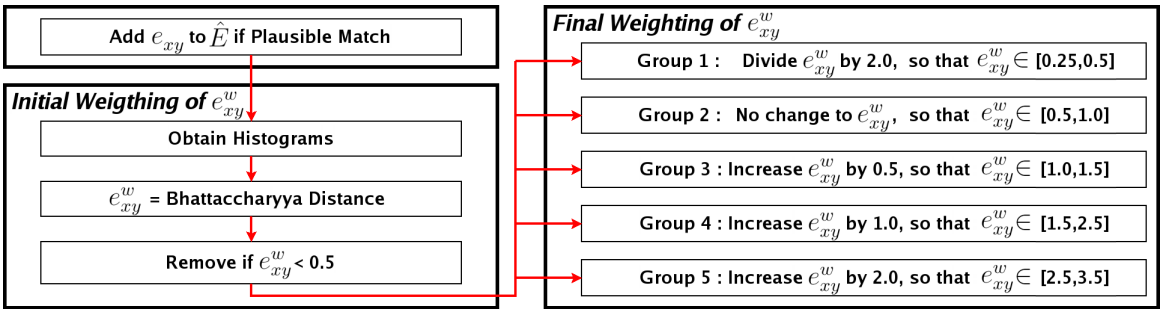


Figure 6.2: Creating and weighting e_{xy}.

created, and each e_{xy} is weighted to indicate the likelihood of a match between p_x and t_y. If there is no likelihood of a match then $e_{xy} \notin \hat{E}$. Figure 6.2 illustrates the process in which a single edge e_{xy} is created and weighted. The creation and weighting of edges in $\hat{E}$ is described in the next section.

In order to obtain the best matching of pedestrians to tracks, a *Maximum Weighted Maximum Cardinality Matching* scheme is employed [212]. In graph theory, a *matching* in $G = (V, \hat{E})$ is a subset, S, of the edges $\hat{E}$ such that no two edges in S share a common end node. A *maximum cardinality matching* has the maximum possible number of edges and a *maximum weighted matching* is such that the sum of the weights of the edges in the matching is maximised. The scheme employed therefore maximises the number of pedestrians matched to tracks, while simultaneously obtaining the maximum weighting for those matches. Figure 6.3 illustrates the matching scheme process, which is described in section 6.2.2. A table of all symbols and thresholds used in this chapter is provided in appendix B.

6.2.1.1 Creating $\hat{E}$

For a correct matching of p_x to t_y, then e_{xy} must be an element of $\hat{E}$ and the weighting of the edge, e^w_{xy}, should be high enough to ensure that e_{xy} is included in the final path determined by the matching scheme. The existence of the edge e_{xy} in the set $\hat{E}$ is determined solely by the constraints of the physical world. For the following three sections, apart from the thresholds set for comparing histograms, all thresholds are determined from observations of pedestrians' 3D physical movements between frames in test sequence data.

To obtain and weight the edges in $\hat{E}$, the following statistics are obtained from each p_x region in frame i;

1. $p_x^{3d^i}$: the position of the centre of mass of a detected pedestrian's 3D head region ortho-

graphically projected onto the groundplane;

2. $p_x^{max^i}$ and $p_x^{min^i}$: the maximum and minimum heights above the groundplane of all the 3D points belonging to the pedestrian that are *visible* in frame i and within the required VOI defined in section 5.2.1;

3. $p_x^{c^i}$: the set of HSV colour values of all foreground points belonging to the pedestrian.

Similarly, all t_y have similar statistics in frame $i-1$; $t_y^{3d^{i-1}}$, $t_y^{max^{i-1}}$, $t_y^{min^{i-1}}$ and $t_y^{c^{i-1}}$. In addition, each t_y has three additional statistics;

1. $t_y^{n^{i-1}}$: the number of frames for which the track has existed;

2. $t_y^{v^{i-1}}$: the velocity of the track in the previous frame, where $t_y^{v^{i-1}} = \left| t_y^{3d^{i-1}} - t_y^{3d^{i-2}} \right| \times \frac{1}{td_{i-2}^{i-1}}$ and td_{i-2}^{i-1} is the time difference (in milliseconds) between frames $i-1$ and $i-2$;

3. $t_y^{3d^i}$: the extrapolated position of the track in the current frame, where if $t_y^{n^{i-1}} < 2$ then $t_y^{3d^i} = t_y^{3d^{i-1}}$, otherwise $t_y^{3d^i} = t_y^{3d^{i-1}} + (t_y^{v^{i-1}} \times td_{i-1}^i)$;

4. $t_y^{s^{i-1}}$: the track state, which is either *walking*, St^w, *accelerating*, St^a, or *standing*, St^s. A person is considered to be walking if they have either; (a) moved in the same direction for 3 consecutive frames, (i.e. the angles between $t_y^{3d^{i-4}}$,$t_y^{3d^{i-3}}$,$t_y^{3d^{i-2}}$ and $t_y^{3d^{i-1}}$ are greater than 90° in each case), or; (b) moved in the same direction for 2 consecutive frames and $dist(t_y^{3d^{i-1}}, t_y^{3d^{i-3}}) > t_{noise}$, where $dist$ is Euclidean distance and t_{noise} is the maximum distance a track's t_y^{3d} is allowed to fluctuate in one frame when they are standing still. In our experiments t_{noise} is set to 0.3 metres, meaning it is expected that a pedestrian's position will fluctuate by up to 0.15 metres from its correct position in any given frame. A person is deemed to be accelerating if $dist(t_y^{3d^{i-1}}, t_y^{3d^{i-2}}) > t_{noise}$. Finally, a person is standing still if neither of the other states are possible.

To determine whether $e_{xy} \in \hat{E}$, p_x is compared to t_y and an evaluation is made whether the match is physically plausible. For example, if the time difference, td_{i-1}^i, between frames $i-1$ and i is one second, and $dist(t_y^{3d^{i-1}}, p_y^{3d^i}) = 20$, where $dist$ is Euclidean distance in metres, the edge e_{xy} is *not* plausible, as the pedestrian p_x would have to moving at a rate of 72km/h! Therefore, if $dist(t_y^{3d^{i-1}}, p_y^{3d^i}) > t_{max}$ then $e_{xy} \notin \hat{E}$, where $t_{max} = dist_{max} \times td_{i-1}^i$ and $dist_{max}$ is the *absolute maximum* distance a pedestrian is assumed to be able to be the walk in a second. In this work, a threshold $t_{avge} = dist_{avge} \times td_{i-1}^i$ is also applied, where $dist_{avge}$ is the *average maximum*

distance a pedestrian is assumed to walk in a second. In our experiments, $dist_{max}$ and $dist_{avge}$ are set to 3 and 2 metres per second respectively (however it should be noted that the minimum value of t_{max} or t_{avge} should be set to t_{noise} regardless of the value of td^i_{i-1}). This limitation of possible pedestrian movement, where $dist(t_y^{3d^{i-1}}, p_y^{3d^i}) > t_{max}$ then $e_{xy} \notin \hat{E}$, defines the first kinematic constraint in this work. As a set, these constraints can be used to remove implausible matches of pedestrians to previous tracks.

A second physical constraint is based on a pedestrian's ability to turn while *walking* at a sufficiently large velocity. It is assumed that due to the forward momentum incurred, a pedestrian can only turn a certain angle θ in a single frame. This constraint can be formulated as follows. Let $\vec{a}$ be the vector from $t_y^{3d^{i-1}}$ to $t_y^{3d^i}$ and $\vec{b}$ be the vector from $t_y^{3d^{i-1}}$ to $p_x^{3d^i}$, and let $\theta = cos^{-1}\left[\frac{\vec{a}.\vec{b}}{|\vec{a}||\vec{b}|}\right]$ be the angle between $\vec{a}$ and $\vec{b}$ (obtained using the dot product). From the statistics of a track it can be assumed that a pedestrian is then moving at a high enough velocity if $dist(t_y^{3d^{i-1}}, p_y^{3d^i}) > t_{noise}$ and either; (a) a person is accelerating very quickly ($t_y^{s^{i-1}} = St^a$ and either $t_y^{v^{i-1}}$ or $t_y^{v^{i-1}}$ is greater than t_{avge}), or; (b) a person is walking and the velocity in the previous frame was greater than that to be expected of a position change due to noise if the pedestrian has suddenly stopped walking ($t_y^{s^{i-1}} = St^w$ and $t_y^{v^i} > \frac{t_{noise}}{2}$). If either of these cases are true then the pedestrian may either stop *or* continue on in roughly the same direction in frame i, i.e. $dist(t_y^{3d^{i-1}}, p_y^{3d^i}) \leq t_{noise}$ *or* $\theta \leq \theta_{max}$, where θ_{max} is the maximum angle that a walking pedestrian can turn per frame. θ_{max} is set to 60° in all our experiments. This is the second kinematic constraints in this work used to remove implausible matches of pedestrians to previous tracks. However, it should be noted that as the time difference, td, between frames increases this constraint is invalidated. In this work it is assumed that the latency between frames is less than the time taken for a pedestrian to stop walking forward and make a turn greater than θ_{max}.

6.2.1.2 Weighting $\hat{E}$

In order to obtain the correct matching of p_x to t_y, a weighting, e^w_{xy}, must be associated with e_{xy}. This value should be high enough to ensure that e_{xy} is included in the final path determined by the matching scheme. As illustrated in figure 6.2, the initial weighting of e_{xy} is assigned by a colour histogram comparison measure between p_x and t_y. This value of e^w_{xy} is then adjusted by a predetermined amount that forces the weights into five distinct groups of varying importance.

In order to obtain the initial value of e^w_{xy}, a normalised histogram for the pedestrian, $p_x^{h^i}$, is created using the hue value from the HSV colour values in $p_x^{c^i}$ using 3D points that lie only in the

overlapping height region between $p_x^{max^i}$ to $p_x^{min^i}$ and $t_y^{max^{i-1}}$ to $t_y^{min^{i-1}}$. A similar histogram, $t_y^{h^{i-1}}$, is created for the track. e_{xy}^w is determined by obtaining the Bhattacharyya distance [213] between the corresponding $p_x^{h^i}$ and $t_y^{h^i}$. This technique results in a value for e_{xy}^w will will lie between 0 and 1. Finally, if $e_{xy}^w < 0.5$ then e_{xy} is removed from $\hat{E}$ as the colour match is deemed to be too weak for a true match between a p_x and t_y to exist.

This weighting, e_{xy}^w, is then altered to force e_{xy} into one of five distinct weighting groups, whereby the *higher* the value of e_{xy}, the *greater* the importance of that weight, and therefore the *greater* the probability of it being chosen for the final matching. These groupings exist in order to reward good matches, established tracks and penalise more implausible, but not impossible, matches. As illustrated in figure 6.2:

- In *Group 1*, the weight is actually decreased by 50% of the original value. This decrease is made in order to discourage plausible but unlikely matches. e_{xy} is part of this group if a person is walking or accelerating ($t_y^{s^{i-1}} = St^w$ or St^a) and either; (a) $dist(t_y^{3d^{i-1}}, p_y^{3d^i}) > t_{avge}$, this discourages the system from attempting to make large jumps in distance, as they rarely occur; (b) $dist(t_y^{3d^{i-1}}, p_y^{3d^i}) > t_{noise}$ and $\theta > \frac{\theta_{max}}{2}$, as if a person is accelerating or walking then these changes in direction are unlikely to occur; or (c) if a pedestrian has walked the same direction for 3 or more consecutive frames and $\theta > \theta_{max}$, even if $dist(t_y^{3d^{i-1}}, p_y^{3d^i}) < t_{noise}$, as the previous history of the track indicates that the angle of the track should be continuous even with respect to noise.

- In *Group 2*, the weighting remains as it is, this is the default group.

- *Group 3* rewards a good match in coverage between p_x to t_y in other areas, besides histograms. So if the overlapping height regions are large (i.e. $\geq 50\%$ overlap), then e_{xy}^w is incremented by 0.5. Note that e_{xy} may be a member of this group *and* groups 4 or 5 at the same time.

- In *Group 4*, a weighting increment is added to e_{xy}^w to ensure that older, more established, tracks have priority to be matched with pedestrians. This ensures that established tracks are not left without a match, while new tracks, which may have been initialised due to noise, have been given a match. This is achieved by determining if $t_y^{3d^i}$ is close to $p_x^{3d^i}$ within a more constrained set of thresholds of angle and distance (simply half the previous thresholds). If this is true then if $t_y^{n^{i-1}} = 2$ then e_{xy}^w is incremented by one.

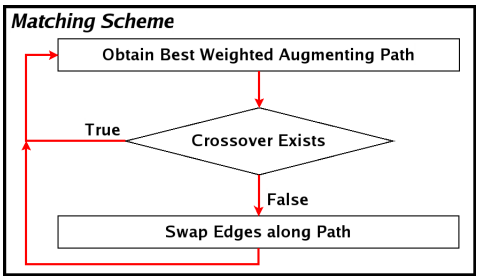


Figure 6.3: Matching scheme.

- In *Group 5*, a similar increment to that in group 4 is added to e^w_{xy}, but if $t^{n^{i-1}}_y > 2$, the weighting is increased by two. Thereby, tracks that have existed for 3 or more frames have priority over those that have existed for 2 frames, and tracks that have existed for 2 or more frames have priority over those that have existed for only 1 frame.

6.2.2 Maximum Weighted Maximum Cardinality Matching Scheme

After $\hat{E}$ has been created and weighted, the matching algorithm is invoked. The matching scheme technique applied in this work – illustrated in figure 6.3 – is based on Berge's Theorem [214], which states that a matching S in G is maximum *iff* there is no augmenting path, P. In graph theory, a *path* is the list of vertices of a graph where each vertex has an edge from it to the next vertex and an *augmenting* path is one with alternating free and matched edges that begins and ends with free vertices. If such a path is discovered then the cardinality of the matching S can be immediately increased by one, simply by switching the edge membership along P. As such, the proposed matching scheme algorithm is initialised with an empty set of matches and then solves the problem by iteratively searching for the augmenting path [212] with the maximum weight. If an augmenting path is found then the edge membership along P is switched. If no augmenting path is found then M is guaranteed to have maximum cardinality with maximum weight, and by traversing through the path the matches of pedestrians to tracks are obtained. This algorithm is a classical solution to the N-to-M association problem using bipartite graphs.

This technique is illustrated in the example of figure 6.4. In this example, there are two detected pedestrians, p_1 and p_2, and three tracks, t_1, t_2 and t_3, see figure 6.4(a). In addition, there are five possible edges, i.e. $\{e_{11}, e_{12}, e_{13}, e_{21}, e_{23}\} \in \hat{E}$, where $e^w_{11} = 2.5$, $e^w_{12} = 1.6$, $e^w_{13} = 0.4$, $e^w_{21} = 2.4$ and $e^w_{23} = 1.4$ – as illustrated in figure 6.4(b). Initially there are no matches in the graph, as shown in figure 6.4(a). The technique iteratively obtains the best augmenting path and uses this path to alter the matching S, resulting in its cardinality increasing by one. It should also

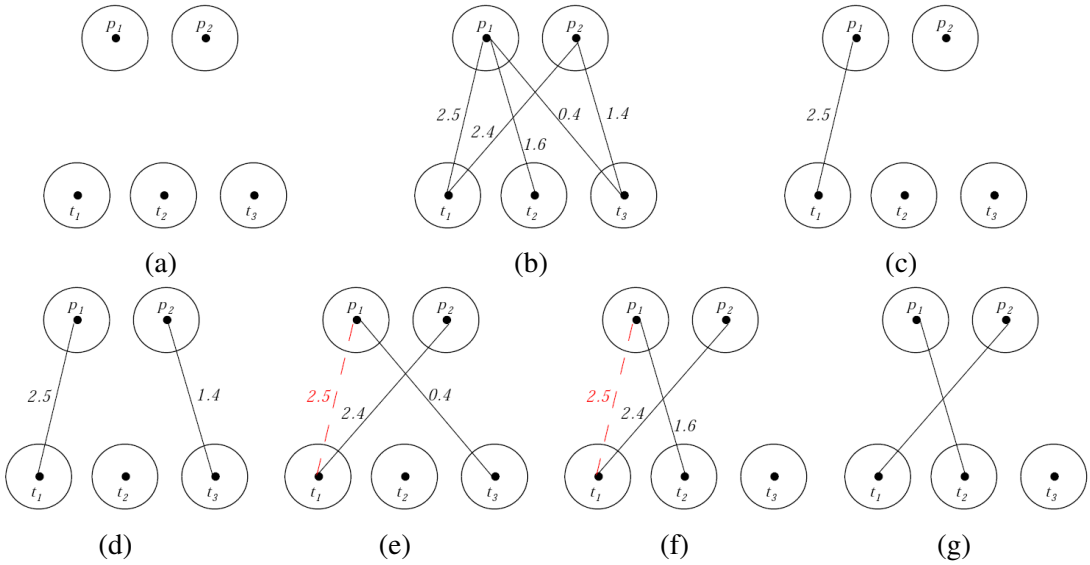


Figure 6.4: Augmenting path; (a) Two detected pedestrians, p_1 and p_2, and three tracks, t_1-t_3; (b) Five possible edges, i.e. $\{e_{11}, e_{12}, e_{13}, e_{21}, e_{23}\} \in \hat{E}$, where $e^w_{11} = 2.5$, $e^w_{12} = 1.6$, $e^w_{13} = 0.4$, $e^w_{21} = 2.4$ and $e^w_{23} = 1.4$; (c) First match; (d) Augmenting path 1, total weight = 1.4; (e) Augmenting path 2, total weight = 0.3 (i.e. $0.4 - 2.5 + 2.4$); (f) Augmenting path 3, total weight = 1.5 (i.e. $1.6 - 2.5 + 2.4$); (g) Final matches.

be noted that a single edge forms an augmenting path if it begins and ends with free vertices. In the example of figure 6.4, the first match is obtained by obtaining the highest augmenting path. As there are no matches already in S this is simplified to obtaining the highest weighted edge in $\hat{E}$, i.e. e_{11}. This path is then added to the matching S, resulting in its cardinality increasing by one, see figure 6.4(c). The next iteration of the algorithm investigates all possible augmenting paths and, if one exists, selects the one with the highest weighting to further increase the cardinality of the matching S. In the example of figure 6.4 three such paths exist. The first – as shown in figure 6.4(d) – can be simply obtained by obtaining the augmenting path of the single edge e_{21}. This path has a weighting of 1.4. The second – depicted in figure 6.4(e) – is an augmenting path consisting of three edges. As shown, the augmenting path has alternating free and matched edges that begin and end with free vertices, i.e. t_3 is free, p_1 is matched, and t_1 is free. In essence the augmenting path matches t_3 to p_1, deletes the match p_1 to t_1 and finally matches t_1 to t_2. The weighting of this path is 0.3, calculated as $0.4 - 2.5 + 2.4$, where 2.5 is subtracted as the match p_1 to t_1 is deleted. Finally, the third augmented match – shown in figure 6.4(f) – matches t_2 to p_1, deletes the match p_1 to t_1 and matches t_1 to t_2. This has the highest weighting of all three augmenting paths at 1.5, calculated as $1.6 - 2.5 + 2.4$. Therefore, this final path is used to update the matching S, resulting in its cardinality increasing by one, see figure 6.4(g). In this simple example there are no more augmenting paths, and as such these are the final matches. However, in the experiments, a number

Figure 6.5: (a) Crossover; (b) Near crossover.

of iterations of the algorithm may be required to obtain the final set of matches for S.

Within the pedestrian tracking module of this work, an alteration to this N-to-M matching scheme algorithm is made that enforces the physical constraints of real-world pedestrian tracking to be taken into account within the matching framework. When creating $\hat{E}$ (see section 6.2.1.1) two kinematic constraints are enforced, which ensures that all single edges $e_{xy} \in \hat{E}$ are physically plausible. However, these constraints do not ensure that *pairs* of edges are physically plausible. Take for example figure 6.5(a), where t_1 is a track traversing the scene from left to right, and t_2 is a second track that is travelling parallel to t_1. In frame i, t_1 or t_2 can be matched to either p_1 or p_2, as each match is physically plausible. However, if t_1 is matched to p_2 and t_2 to p_1, then t_1 and t_2 must pass through, or *crossover*, the same physical space between frames $i-1$ and i. Depending on the time difference between the two frames this may be physically impossible, as is the case in our experiments, where the latency between frames difference is typically less than half a second. As such, pairs of edges of this type are not allowed to coexist in a legitimate matching. As such, a constraint is imposed that in a legitimate matching, no two *physical* track segments between frames i and $i-1$ may be within a distance of 10 cm of each other. This eliminates all crossovers, and near crossovers, such as that in figure 6.5(b), where $t_1^n = 1$ and $p_1^{3d^i} \approx t_1^{3d^{i-1}}$, where although no actual crossover has occurred, the two track segments again at some stage between i and $i-1$ occupy the same physical space. This limitation of possible pedestrian movement defines the third, and final, kinematic constraint applied in this work.

6.2.3 Pedestrian Detection Rollback Loops

After pedestrians have been assigned to tracks, rollback loops are used to backtrack the pedestrian detection module in an attempt to find lost tracks, and thus further reduce over and under-segmentation. The rollback scheme employs three separate loops (see figure 6.6); the first two aim to locate all lost tracks using two different techniques, the second of which also reduces under-segmentation; the third is designed to reduce over-segmentation.

If t_y is unmatched then the first rollback loop attempts to find the missing pedestrian, p_x, in the

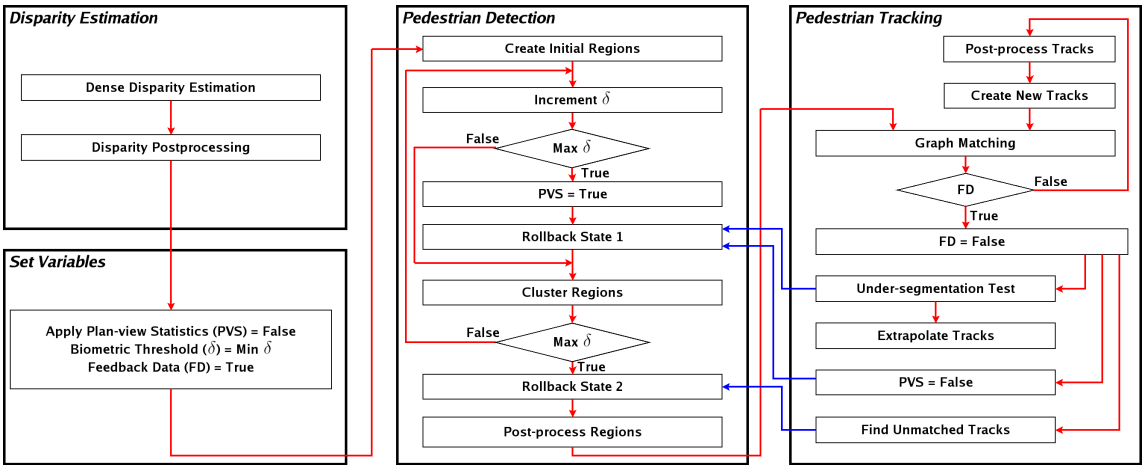


Figure 6.6: Detailed system overview.

current frame. The post-processing stage of the pedestrian detection module (see section 5.3.4) declares a region as noise if it falls below certain thresholds, such as the minimum number of pixels or if it does not span a pre-defined range of heights. However, scenarios such as severe occlusion could force p_x below these thresholds. To retain this lost region, the tracking module backtracks the pedestrian detection module to just before post-processing occurs and reapplies post-processing at half the original thresholds. If any *new* regions emerge that may be a feasible continuation of the lost track, then the weighted bipartite matching scheme is reiterated. If t_y becomes matched then that new pedestrian region remains and all other regions added by this module are removed. An example of this rollback loop can be seen in figures 6.7(a) and (b). In figure 6.7(a) there are no rollback loops and one pedestrian (at the right hand side) is not detected due to occlusion and a lack of foreground gradient information. By applying this first rollback loop the person can be correctly segmented and the track can continue – see the blue track in figure 6.7(b). Note that in the plan-view images (i.e. row 2) of these figures, the white lines indicate the bounds of the scene (which are defined with respect to the visible groundplane within the scene), the position of detected pedestrians in that frame are illustrated by a circle of the same colour as their bounding box in row 1, and tracks are depicted as "tails" from the centre of the circle to previous positions in the scene.

If t_y is still unmatched, a second rollback loop is employed that makes the assumption that under-segmentation of pedestrians occurred resulting in two tracks, t_1 and t_2, competing for the same region, p_1. The rollback loop backtracks the pedestrian module to before the final iteration of region clustering, which is then skipped and the regions are post-processed. If p_1 has been segmented into 2 distinct regions, p_1a and p_1b, where the orientation of $t_1^{3d^i}$ to $t_2^{3d^i}$ is similar to

that of $p_1a^{3d^i}$ to $p_1b^{3d^i}$ then a possible match may exist. In this approach, the maximum difference in orientation is set at $\pm 22.5°$, therefore allowing a total range in orientation difference of $45°$. It is believed that this value allows enough variation in orientation, while simultaneously avoiding the case of incorrect matches from the rollback loop. As in the first rollback loop, if the re-segmentation is successful the weighted bipartite matching scheme is reiterated. However, if the re-segmentation is not possible but an attempt was made, i.e. p_1 exists whereby it can be matched to either t_1 and t_2 but it could not be segmented into two regions, then it is assumed that p_1 actually contains 2 pedestrians, and the unmatched track is extrapolated. An example of this rollback loop can be seen in figures 6.7(c) and (d). In figure 6.7(c) there are no rollback loops and under-segmentation occurs (notice how two pedestrians are clustered together into one region) within the light blue bounding box. By applying this second rollback loop the two pedestrians can be correctly segmented and their respective tracks can continue – see figure 6.7(d).

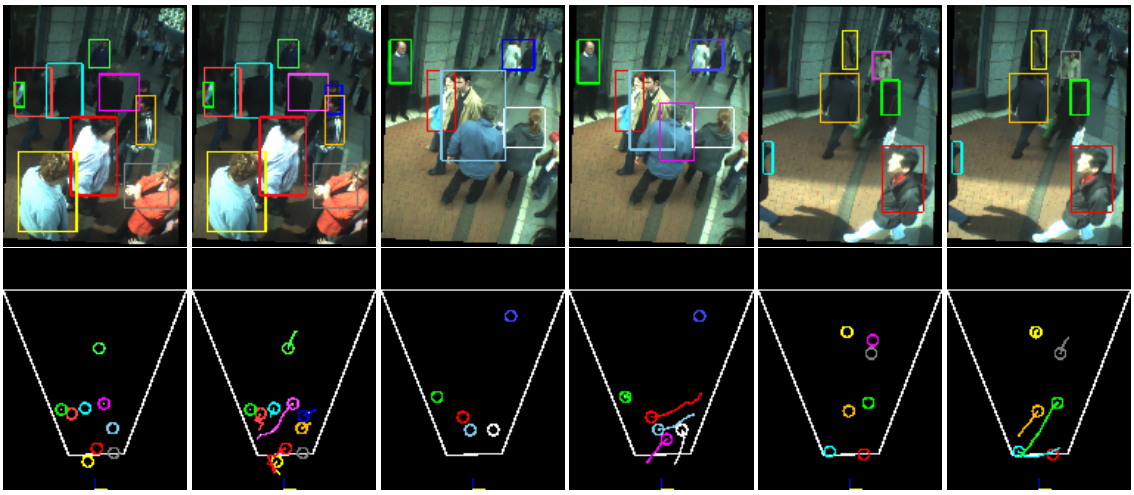
The third rollback loop is designed to reduce over-segmentation by examining all *unmatched* pedestrians. The pedestrian detection module is backtracked to before the final stage of merging regions and the under-segmentation test is turned off. The final clustering stage and post-processing is re-iterated and it is determined whether the unmatched pedestrian region has become merged with a second region. If it does, then the region is considered to be over-segmented and two regions remain merged. An example of this final rollback loop can be seen in figures 6.7(e) and (f). In figure 6.7(e) there are no rollback loops and over-segmentation of a pedestrian towards the back of the scene occurs. By applying this third rollback loop the over-segmentation can be correctly removed – see the grey track in figure 6.7(f).

6.2.4 Track Post-processing

The final stage of the tracking framework is to post-process tracks with a view to increasing track stability with respect to pedestrian over-segmentation and occlusion problems. In order to increase robustness against these issues the global relationships between tracks must be examined.

6.2.4.1 Pedestrian Over-segmentation

If the tracked pedestrian t_1 is over-segmented in frame i as two regions p_1a and p_1b, then a choice has to be made whether to match t_1 to p_1a or p_1b. Let t_1 choose p_1a and let t_1b be the new track initiated by p_1b. Each separate choice affects t_1's statistics in frame $i+1$, meaning that a bad

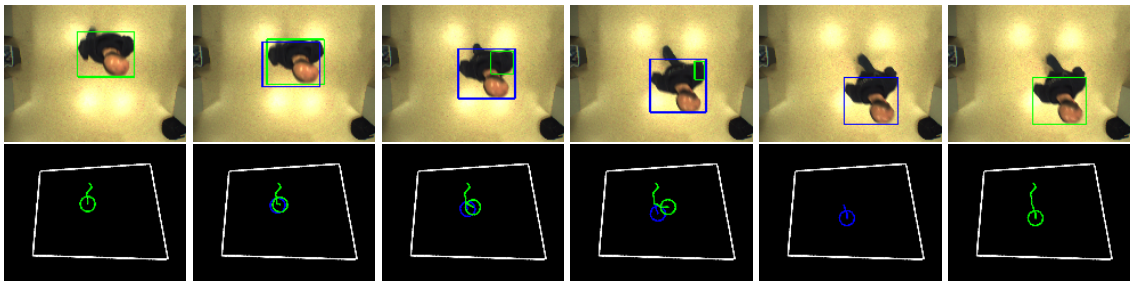


(a) (b) (c) (d) (e) (f)

Figure 6.7: Rollback loops; (a) Lost pedestrian without rollback loop 1; (b) Pedestrian detection with rollback loop 1; (c) Under-segmented pedestrian without rollback loop 2; (d) Pedestrian detection with rollback loop 2; (e) Over-segmented pedestrian without rollback loop 3; (f) Pedestrian detection with rollback loop 3.

choice could end the track prematurely, however the new track from the choice not taken may still exist. If this is the case, then the wrong choice was made. An example of this scenario can be seen in figure 6.8. In figure 6.8(b) the pedestrian is over-segmented and the single previous track (i.e the green track) has a choice of matches due to the over-segmentation. In this scenario the track chooses one of the two as a match and a new track (i.e. the blue track) is created for the second region. In general, if this scenario occurs and the pedestrian is *correctly* segmented in the following image the longer track (i.e. the green track) would take precedence and the track would continue on correctly. However, occasionally the continuation of the green track may not be possible due to changes in the track's statistics. For example, the choice of match in the previous frame ensures that the angle between the direction of travel of the track and the correct continuation of the track becomes greater than θ_{max}. An example of this scenario is shown in figures 6.8(c)-(e) where due to successive over-segmentations the green track is led off course towards the right, therefore in figure 6.8(e) a match between the last position of the green track and the single pedestrian is not possible, leaving the green track to be terminated and the blue track to continue onwards. This type of scenario is clearly undesirable.

This type of occurrence can be rectified by flagging possible over-segmentations and the resultant choice in frame i. This is achieved by flagging all *new* tracks in frame i that are within $1.5 \times |\text{lo}|$ of *older* tracks (where $|\text{lo}|$ is the width of the shoulders of a pedestrian from an older track in frame i using the biometric model of section 5.3.1). Then if t_1 is discontinued before $t_1 b$



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

Figure 6.8: *Oversegmentation* issues. Frame numbers; (a) 149; (b) 150; (c) 151; (d) 152; (e) 153 without post-processed tracks; (f) 153 with post-processed tracks.

and the two separate tracks have not diverged (greater than $1.5 \times |\text{lo}|$ in distance) *or* t_1b has not demonstrated that it is a stable track by being able to reach a walking state, t_1 is allowed to "steal" the track of t_1b. If this scenario occurs, the history of t_1 is replaced by that of t_1b for the duration of t_1b's lifespan. The result of this post-processing step can be seen in figure 6.8(f) where the blue track is recognised as a possible over-segmentation of the green track, and therefore as the green track terminates before the blue track (and the blue track has demonstrated that it is stable track) the history of the blue track is "stolen" by the older track.

6.2.4.2 Explicit Occlusion Analysis

The second and final post-processing stage is used to detect occluded pedestrians in the current frame. These are either; (1) pedestrians that have not been detected via the pedestrian detection module or any of the rollback loop mechanisms (i.e. they are completely occluded), or (2) pedestrians that have been detected and are significantly occluded (i.e. they are partially occluded). The first type of detection (i.e. of completely occluded pedestrians) is intended to maintain tracks through momentary, but complete, pedestrian occlusions, therefore leading to the correct continuation of a track. The reason for detecting partial occlusions is to determine if the head region of the pedestrian is occluded. If this is the case the 3D position of the pedestrian (and therefore the direction of travel of the track) in the scene may become unreliable and therefore should be flagged as such.

In order to detect if a tracked pedestrian, t_1, is occluded in frame i, the position of the pedestrian is compared to the positions of all other pedestrians detected in frame i that are *closer* to the camera than t_1. An overview of this process is illustrated in figure 6.9. Let t_1 be matched to p_1 in frame i and therefore the position of the track in this frame is given by $p_1^{3d^i}$ (note: if t_1 is not matched in frame i, i.e. the track is completely occluded, then the predicted position of the track,

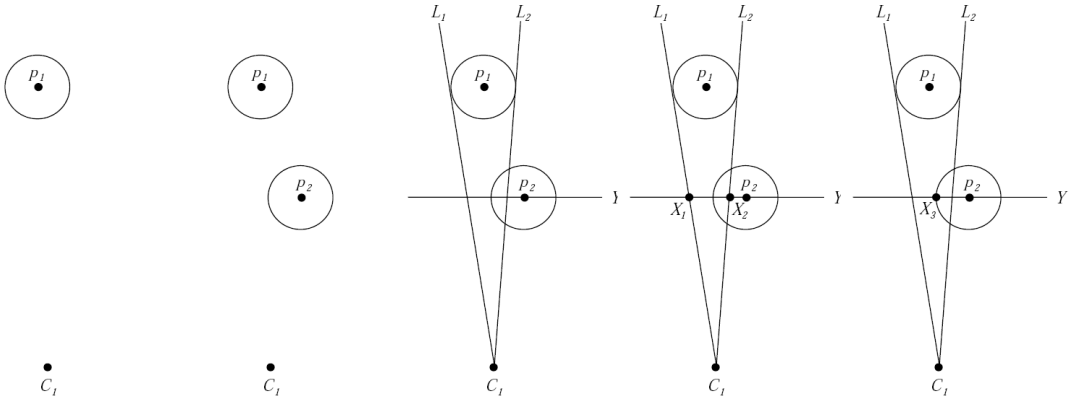


Figure 6.9: Occlusion analysis; (a) Pedestrian p_1; (b) Pedestrian p_2; (c) L_1 and L_2; (d) X_1 and X_2; (e) X_3.

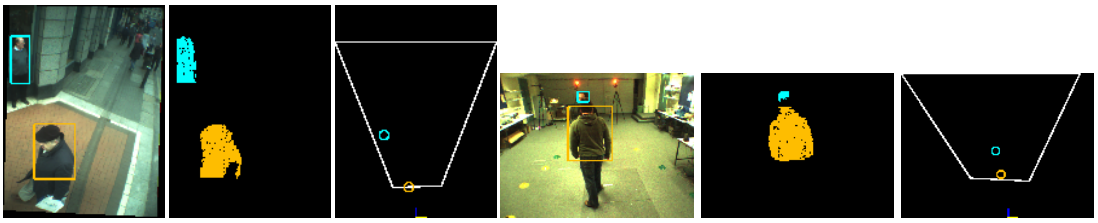
$t_1^{3d^i}$, is applied). In figure 6.9(a) the circle around p_1 is of diameter $|\text{lo}| \times t_1^{max^i}$ (where $|\text{lo}|$ is the width of the shoulders using the biometric model of section 5.3.1) and C_1 is the camera centre of camera 1 and the world Euclidean origin. If a second pedestrian, p_2, exists that is closer to the camera than p_1, then it is possible that p_2 occludes p_1, see figure 6.9(b), where the circle around p_2 is of diameter $|\text{lo}| \times p_2^{max^i}$ (or $|\text{lo}| \times t_x^{max^i}$ if p_2 is matched to the track t_x).

To determine if p_2 occludes p_1 three lines are drawn, see figure 6.9(c); L_1, which passes through C_1 and is a tangent to the left side of the circle around p_1; L_2, which passes through C_1 and is a tangent to the right side of the circle around p_1; and Y, which is parallel to the x-axis of the world co-ordinate system and passes through the position of p_2 on the groundplane. Using these lines two points are determined, see figure 6.9(d); X_1, which is the intersection point between L_1 and Y; and X_2, which is the intersection point between L_2 and Y. In addition, a third point, X_3, which is closest point on the circle $|\text{lo}| \times p_2^{max^i}$ around p_2 to either X_1 or X_2 (whichever is the farthest from the centre of the pedestrian p_2), see figure 6.9(e).

If p_2 is to occlude p_1 then X_3 must lie between X_1 and X_2, and if this is the case then the percentage of overlap can be determined via

$$Occlusion\% = \frac{dist(X_3, X_2)}{dist(X_1, X_2)} \times \frac{100}{1} \tag{6.1}$$

In this work p_2 *may* be determined to occlude p_1 if this percentage is greater than 50%, which would therefore cover more than half the occluded pedestrians head region. However, a plan-view analysis on its own is not sufficient to determine if occlusions occur. This point is illustrated in figure 6.10, where two different scenarios are presented. In each of these scenarios the plan-view



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

Figure 6.10: Occlusion in the rectified image; (a)-(c) *Grafton* sequence; (d)-(f) *Vicon* sequence.

images, figures 6.10(c) and (f), indicate a positive occlusion of the farther pedestrian according to the proposed criterion. However, from figures 6.10(a) and (b) it is clear than occlusion does *not* in fact occur for the first scenario. Therefore, for occlusions to occur an overlap of regions in *both* plan-view and *rectified images* must be present.

To determine if p_1 is occluded by p_2 in the rectified image, a bounding box is obtained around the pedestrian p_1's head region and a second bounding box is obtained around the whole of pedestrian p_2's body region. If the two bounding boxes overlap then it is deemed that p_1 is occluded by p_2. If there is an overlap in *both* plan-view and rectified images then an occlusion is declared. Using this explicit occlusion detection, pedestrians can be extrapolated through momentary complete occlusions.

A similar technique is applied for partially occluded pedestrians, except that only pedestrians who have their heads occluded are detected, as the centre of mass for the head region is used as the tracking position of the pedestrian. If the head is fully or partially occluded in frame i then the 3D position of the pedestrian (and therefore the direction of travel of the track) in the scene may be unreliable. A pedestrian is only deemed to be partially occluded if the height of the detected pedestrian in frame i is less than the height of the body in previous track positions minus the head region (i.e. if $p_x^{max^i} < t_x^{max^i} - (p_x^{max^i} \times |\text{af}|)$, where $|\text{af}|$ is the biometric distance from the top of the head to the base of the skull). If this is the case then in frame $i+1$ the second physical constraint of section 6.2.1.1, which based on a pedestrian's ability to turn while walking at a high enough velocity, is ignored for that track. This allows tracks to become more robust in the presence of partial occlusions.

6.2.4.3 Update Track Statistics

The final stage of the process is to extrapolate tracks if possible, to remove lost tracks and to create new tracks for all unmatched pedestrians in the current frame. If a track has a match in the current

frame then the tracks statistics (such as track state, position and colour models are updated). The colour model is simply updated by replacing all relevant foreground pixels in the track HSV colour model with those within the height bounds of the matched pedestrian (i.e. replace all pixels from $t_x^{c^{i-1}}$ within the bounds of $p_x^{max^i}$ and $p_x^{min^i}$ with those pixels from $p_x^{c^i}$). Finally, all tracks that require to be extrapolated are extended as long as the head region of the extrapolated position of the pedestrian is within the rectified image bounds and within the 3D volume of interest. In our experiments, a track is only extrapolated for a maximum of 2 frames, after which if a match for the track can still not be found then the track is removed. In addition, a track is only extrapolated if it is a well established track, i.e. it has existed for over 2 frames, and therefore unlikely to be caused by noise or over-segmentation.

6.3 Experimental Results

We quantitatively evaluated the proposed pedestrian tracking technique using the same challenging datasets as in chapter 5. The tracking module can be split into two distinct areas of evaluation. The first is to evaluate the effect of rollback loops and occlusion analysis with regards to pedestrian detection precision, recall and 3D statistics. The second is to evaluate the matching process with regards to the creation, removal and continuation of tracks and the suppression of non-pedestrian tracks. Finally, in section 6.3.3 an overview of the final pedestrian detection and tracking system is presented with the use of the illustrative results.

6.3.1 Pedestrian Detection Evaluation with Tracking Rollback Loops

In order to evaluate the effect of the rollback loops and the explicit occlusion analysis techniques the same two methodologies for matching detected to groundtruth pedestrians were applied as in chapter 5. The first technique (see section 5.4.1) evaluates the proposed approach via 2D image plane comparison techniques. The second technique (described in section 5.4.2) evaluates the proposed approach via Vicon 3D groundtruth information. One minor change is made to the 3D evaluation. Previously, using the biometric model a value of $|\text{lo}|$ (i.e. the width of the shoulders) was determined for each groundtruth person's position. If the proposed technique detects a person within a Euclidean distance of this value $|\text{lo}|$ to the groundtruth position then a match was made between the detected and groundtruth pedestrian. In this evaluation, due to extrapolation, this difference was allowed to be greater than the value of $|\text{lo}|$ for a maximum of 2 frames as long as

this distance then returned to less than |lo| is subsequent frames. This change was made to ensure that extrapolated positions of pedestrians were not punished unnecessarily if their position was momentarily incorrect.

The precision and recall values are presented in tables 6.1, 6.2 and 6.3. In these tables two values, A_B, are provided for each precision and recall value. A represents the precision or recall value for the pedestrian detection module, which incorporates rollback loops from the pedestrian tacking module. B represents the percentage difference of the precision or recall value with respect to the pedestrian detection module *without* rollback loops as seen in tables 5.4, 5.8 and 5.12 of chapter 5.

From these results it can be seen that the precision on *every* sequence is improved with a minimum increase of 0.56% for the *Vicon* 2 sequence to a maximum increase of 9.03% for the *Overhead* sequence. Using the rollback mechanisms, the precision of every sequence is raised to over 91%. The increase for precision is generally greater in the 2D sequences (obtaining an average increase of 5.61%) than in the 3D *Vicon* sequences (obtaining an average increase of 1.16%). This is to be expected as the pedestrian detection module (without tracking rollback loops) precision on the 2D sequences is on average 9.58% lower than those of the 3D sequences. Many of the false-positive detections appear for only one or two frames and as such cannot be tracked by the system for the minimum required number of frames for them to be labelled as valid pedestrian regions. It is therefore expected that sequences with more false-positives (and hence lower precision) should benefit most from this feature. This is shown to be the case, whereby sequences with lower precision values tend to record higher increases.

However, improvements in recall are generally greater in the 3D sequences (obtaining an average increase of 3.96%) than in the 2D sequences (obtaining an average *decrease* of 0.33%) – in fact only the *Corridor* sequence recorded an increase of recall (of 3.12%) with the rollback loops. This increase occurs despite the fact that the recall of the 3D sequences without rollback loops was already on average 5.38% higher than that of the 2D sequences. The reason for this increase in the 3D sequences lies in *why* pedestrians were originally missed by the pedestrian detection module (see tables 5.6 and 5.9 of chapter 5). Due to the position and orientation of the camera, the number of pedestrians missed due to *occlusions* is significantly higher in the 3D *Vicon* sequences (on average 85.3% of all missed pedestrians – see table 5.9 of chapter 5) and the *Corridor* sequence (43.75% of all missed pedestrians – see table 5.6 of chapter 5) than any other sequence (the next highest is 22.77% from the *Grafton* 1 sequence). Many of these are momentary

occlusions, whereby the pedestrian is occluded for just 1 or 2 frames, and as such the proposed technique can track such pedestrians through the occlusions, significantly reducing the number of pedestrians lost to occlusion. This therefore leads to an increase in recall.

For the other sequences the main reason for missed pedestrians are due to "other reasons" (on average 68.89% of missed pedestrians for the 2D sequences). These "other reasons" (as outlined in section 5.4.1) are usually due to (1) a lack of accurate disparity or foreground edge information resulting in the region being removed as background or (2) a pedestrian appearing at the edge of the scene of one image and not appearing in the opposite image. However, there only exists a rollback loop to address inaccurate foreground edge information. Therefore, pedestrians missed due to inaccurate or missing disparity information are not addressed by the rollback loops.

In addition, for a region to call a rollback loop it must have existed for a minimum of 3 frames. This is designed to deny rollback loop access to tracks that may have been caused by noise. However, (as outlined in section 5.3.3) as disparity accuracy degrades with respect to distance from the camera, the farther a pedestrian is from the camera the greater the likelihood that a pedestrian may be missed or under-segmented. Therefore, for the continuation of valid pedestrian tracks at large distances from the camera, the access to certain rollback loops are more likely to be required. These, however, are denied to *new* pedestrian tracks until they have been tracked for the minimum required number of frames. Therefore, some valid pedestrian tracks are removed as noise if in the first 3 frames; (a) the pedestrian is under-segmented; (b) the pedestrian is missed; (c) the pedestrian becomes occluded; or (d) the pedestrian enters and exits the scene. In addition, if pedestrians are correctly detected there exists the possibility (which increases in likelihood with distance from the camera) that the 3D position of the pedestrian is not accurate. If this occurs then the possibility of a correct match failing to pass the kinematic constraints increases. If the kinematic constraints are not passed then the correct continuation of the track becomes impossible. This effect can lead an increase in the number of valid pedestrian tracks that are removed as noise, especially when pedestrians enter the scene at a large distance from the camera.

In four of the five 2D sequences, the number of correct pedestrians removed as noise is greater than the additional number of pedestrians obtained via the rollback mechanisms. Therefore, these sequences have reduced recall values. This could be simply countered by reducing the number of frames that a possible pedestrian region must be tracked before being considered as valid. However, this would result in a reduction in precision values. In all of the sequences tested, the *Grafton* 3 sequence has the greatest drop in recall. As outlined in section 5.4.1, this sequence – see

figure 6.14 – suffers from a lack of foreground gradient information due to strong cast shadows. Therefore, pedestrians in shadows are more likely to be removed as background. This therefore results in a large drop in recall due to fewer tracks existing for the minimum number of 3 frames, and more tracks being terminated. However, the reduction of 3.60% in recall is coupled with an increase of 7.90% in precision. This increase in precision is mirrored in all other sequences that exhibit a decrease in recall. For the *Grafton* 2, *Grafton* 1 and *Corridor* sequences there is an increase in precision equivalent to 3.11, 4.89 and 11.86 times the decrease in recall respectively. Therefore, removing tracks that exist for less than the minimum number of 3 frames can viewed as a successful system feature. Overall, the rollback loops and explicit occlusion analysis techniques provide an average increase of 2.93% and 2.35% in precision and recall values. These increases result in final precision and recall values of 97.82% and 91.49% for the 10 challenging sequences. These results point to extremely accurate performance of the proposed approach under a variety of scenarios, camera positions, lighting conditions and pedestrian densities.

In addition to precision and recall values, the 3D statistics of each tracked pedestrian in the 3D *Vicon* sequences were compared to the corresponding values without the rollback loops (see tables 5.10 and 5.11 of chapter 5). These comparisons are presented in tables 6.4 and 6.5. For these figures it can be seen that there is a minor overall average increase in both height and positioning error of 0.16cm and 0.31cm respectively. These values present an average decrease in overall accuracy of 0.05% and 0.18% in height and positioning statistics respectively. These increases mainly arise due to the extrapolation of occluded pedestrian positions. However, the increases are not significant when the maximum estimated error between a pedestrian's detected and real-world position between two frames (i.e. t_{noise}) is set to 30cm. From these results it is shown that these 3D statistics remain highly accurate with respect to the number of pedestrians and occlusions within the scene.

6.3.2 Pedestrian Tracking Evaluation

The second stage of the evaluation is focused on the proposed pedestrian tracking technique. This evaluation can be spit into two main sections. The first evaluates the tracks as a whole whilst the second evaluates the bipartite matching scheme with regard to correct and incorrect matches.

Sequence	*Groundtruth*	*Detected*	*Correct*	*Precision*		*Recall*	
Grafton 1	666	577	558	96.71	+5.13	83.78	-1.05
Grafton 2	754	671	652	97.17	+4.95	86.47	-1.59
Grafton 3	555	446	420	94.17	+7.90	75.68	-3.60
Grafton Total	1975	1694	1630	96.22	+5.86	82.53	-1.98
Overhead Total	657	641	610	95.16	+9.03	92.85	-0.76
Corridor Total	1027	969	883	91.12	+2.94	85.98	+3.12
2D Total	3659	3304	3123	94.52	+5.61	85.35	-0.33

Table 6.1: 2D sequences pedestrian detection evaluation with rollback loops.

Sequence	*Groundtruth*	*Detected*	*Correct*	*Precision*		*Recall*	
Vicon 1	198	198	198	100.0	+3.41	100.0	0.00
Vicon 2	526	533	526	98.69	+0.56	100.0	+0.57
Vicon 4	1296	1301	1291	99.23	+1.50	99.61	+3.32
Vicon 8_A	2104	1980	1976	99.80	+1.11	93.92	+4.85
Vicon 8_B	2120	1942	1942	100.0	+0.86	91.60	+4.67
3D Total	6244	5954	5933	99.65	+1.16	95.02	+3.96

Table 6.2: 3D sequences pedestrian detection evaluation with rollback loops.

Sequence	*Groundtruth*	*Detected*	*Correct*	*Precision*		*Recall*	
Synthetic Total	97	95	93	97.89	N/A	95.88	N/A
2D Total	3659	3304	3123	94.52	+5.61	85.35	-0.33
3D Total	6244	5954	5933	99.65	+1.16	95.02	+3.96
Total	10000	9353	9149	97.82	+2.93	91.49	+2.35

Table 6.3: Evaluation of all sequences with rollback loops. Two figures, A $_{\text{B}}$, are provided for each of the Precision and Recall columns in tables 6.1, 6.2 and 6.3. A represents the precision or recall value for the pedestrian detection module, with rollback loops from the pedestrian tacking module, $_{\text{B}}$ represents the percentage difference of the precision or recall value with respect to the pedestrian detection module without rollback loops. For example, in table 6.1 for row 2 column 5, the *Grafton* 1 sequence records a precision of 96.71%, which is +5.13 to the precision value of the pedestrian detection module without rollback loops as given in table 5.4. Note: The *Synthetic* results in table 6.3 are entered for consistency reasons between tables 6.3 and 5.12, and N/A represents "Non applicable" as *Synthetic* dataset consists solely of images and not sequences. Therefore, no tracking can be implemented and the resultant values have not changed.

Sequence	*Av. Error*	*Av. % Error*
Vicon 1	10.45 -0.31	2.73 -0.08
Vicon 2	11.70 -0.05	3.27 -0.02
Vicon 4	8.17 +0.06	2.29 +0.02
Vicon 8_A	10.03 +0.22	2.85 +0.06
Vicon 8_B	7.44 +0.30	2.05 +0.08
Total	8.94 +0.16	2.5 +0.05

Table 6.4: 3D sequences distance results with rollback loops (in cm).

Sequence	*Av. Error*	*Av. % Error*
Vicon 1	7.26 -0.01	4.01 -0.01
Vicon 2	8.35 -0.05	4.60 -0.03
Vicon 4	9.79 +0.27	5.39 +0.15
Vicon 8_A	10.37 +0.40	5.84 +0.23
Vicon 8_B	10.55 +0.33	5.93 +0.18
Total	10.02 +0.31	5.60 +0.18

Table 6.5: 3D sequences height results with rollback loops (in cm). Two figures, A ʙ , are provided for columns 2 and 3 in tables 6.4 and 6.5. A represents the absolute average error or average error percentage for the pedestrian detection module with rollback loops from the pedestrian tacking module, ʙ represents the difference in the corresponding for the pedestrian detection module without rollback loops. For example, in table 6.4 for row 2 column 2, the *Vicon* 1 sequence records an average error in positioning pedestrians of 10.45cm, which is 0.31cm better in accuracy than the corresponding value in table 5.10.

6.3.2.1 Track Evaluation

For the first stage of the pedestrian tracking evaluation each track within the 10 test sequences were classified into one of four groups. A *Correct* track is one that is initiated by a single pedestrian and throughout its lifetime incorporates the same single pedestrian. This type of track is defined as *Correct* as it behaves exactly as designed. In some cases, a pedestrian track may be incorrectly terminated in frame i, but in frame $i + n$ the same pedestrian is then assigned a new track. If this event occurs then the second track is defined as an *Additional* track, as the pedestrian should have only one track. In our experiments, a pedestrian track is correctly terminated if the pedestrian exits the bounds of the volume of interest of the scene *or* the person becomes occluded for three consecutive frames. The third case is a track caused by the *Over-segmentation* of a pedestrian, where a pedestrian is segmented into two or more distinct regions whereby each region has a separate track. In this case the oldest track is defined as the *Correct* track and any additional tracks are defined as over-segmented. The final case is a track caused by some *Background* object. An analysis in these terms of tracking results are presented in tables 6.6, 6.7 and 6.8 .

The values provide an evaluation at a higher level of abstraction than precision and recall val-

ues. However, system evaluation from the tables alone can be somewhat misleading. For example, in table 6.7 it can be seen that only 60% the *Vicon* 4 sequence tracks are *Correct* as 40% occur due to over-segmented regions. This information on its own indicates poor performance, however, using the precision (99.23%) and recall (99.61%) information reveals that the 4 incorrect tracks are caused by just 10 pedestrian over-segmentations out of 1301 detected regions. In general, by applying the information of these tables in conjunction with the precision and recall values a highly detailed view of the system can be obtained. For example, the cause of the 1.31% loss in precision of *Vicon* is due to 7 additional false-positives (see table 6.2), which according to table 6.7 is due to 2 tracks caused by the over-segmentation of pedestrians.

The majority of *Additional* tracks are caused by similar reasons that led to a decrease in precision in section 6.3.1. As in that scenario, *Additional* tracks tend to be caused when operating towards the maximum detection distance from the camera. At these distances the 3D position of a correctly detected pedestrian tends to become susceptible to noise and as such can cause inaccurate track position and prediction. This can lead to the premature removal of a track as the correct continuation of the track may become impossible due to the kinematic constraints imposed upon pedestrian-to-track matches (see section 6.2.1.1). In this case, the previous track is terminated and a new track is created. In our experiments (not shown in the tables), it was determined that 21 (or 75%) of all *Additional* tracks were caused by this effect – 7 (or 25%) were caused by the inability to segment a pedestrian for the continuation of a track in frame i, but the pedestrian was then correctly segmented in frame $i + 1 + n$, where $n \geq 0$. Due to the majority of *Additional* tracks occurring when a pedestrian is entering or exiting the scene the majority of these tracks (or the initial *Correct* correct track) are small in length. By increasing the thresholds on the kinematic constraints, many of the *Additional* tracks could be eliminated. However, in these sequences this may lead to an increase in the number of incorrect matches in the bipartite matching scheme. However, an increase in these values may be beneficial if the sequence frame rate could also be increased. Therefore, due to the low value of td, a higher value of $dist_{avge}$ and $dist_{max}$ could be tolerated. In addition, a fourth possible pedestrian track state, *running*, St^r, could be introduced.

Finally it can be seen that only 1.68% of all tracks were caused by background regions. These tracks occur in two separate sequences. The first is in the *Grafton* 3 sequence where a background track is caused by the reflection of a pedestrian (who is not in the cameras field of view) on a window in the scene – see appendix C for more information on this event. The other 4 regions occur in the *Corridor* sequence and are due to changes in global illumination on the pillar causing

Sequence	*Total Tracks*	*Correct*		*Additional*		*Overseg.*		*Background*	
Grafton 1	72	63	87.50	6	8.33	3	4.17	0	0.00
Grafton 2	60	55	91.67	3	5.00	2	3.33	0	0.00
Grafton 3	50	41	82.00	2	4.00	6	12.00	1	2.00
Grafton Total	182	159	87.36	11	6.04	11	6.04	1	0.55
Overhead Total	74	30	73.17	4	9.76	7	17.07	0	0.00
Corridor Total	41	51	68.92	8	10.81	11	14.86	4	5.41
Total	297	240	80.81	23	7.74	29	9.76	5	1.68

Table 6.6: 2D sequences pedestrians tracking evaluation overview.

Sequence	*Total Tracks*	*Correct*		*Additional*		*Overseg.*		*Background*	
Vicon 1	1	1	100.0	0	0.00	0	0.00	0	0.00
Vicon 2	3	2	66.67	0	0.00	1	33.33	0	0.00
Vicon 4	10	6	60.00	0	0.00	4	40.00	0	0.00
Vicon 8_A	38	34	89.47	3	7.89	1	2.63	0	0.00
Vicon 8_B	41	39	95.12	2	4.88	0	0.00	0	0.00
Total	93	82	88.17	5	5.38	6	6.45	0	0.00

Table 6.7: 3D sequences pedestrians tracking evaluation overview.

Sequence	*Total Tracks*	*Correct*		*Additional*		*Overseg.*		*Background*	
2D Total	297	240	80.81	23	7.74	29	9.76	5	1.68
3D Total	93	82	88.17	5	5.38	6	6.45	0	0
Total	390	322	82.56	28	7.18	35	8.97	5	1.28

Table 6.8: All sequences pedestrians tracking evaluation overview. Two figures, A B, are provided for columns 2-5 in tables 6.6, 6.6 and 6.8. A represents the total number of tracks classified as either correct, additional, over-segmented or background, B represents the percentage of A with respect to the total number of tracks for that sequence. For example, in table 6.6 for row 2 column 3, 63 tracks were classified as correct, which equates to 87.50% of the 72 total number of tracks in the *Grafton* 1 sequence.

significant foreground gradient edge regions, and therefore foreground regions. Two of these tracks can be seen in figures 6.12(a) and 6.12(d). For all three scenarios (i.e. *Additional*, *Overseg.* and *Background*) the number of incorrect regions could be reduced by increasing the minimum number of frames that a track must exist before it is considered as being a valid foreground region. This increase would, however, result in an overall decrease in recall values for the proposed system as in some sequences, especially the *Grafton* sequences, some pedestrians only appear for a small number of frames.

6.3.2.2 Bipartite Match Evaluation

The second evaluation of tracking focuses on the bipartite matching scheme. In these experiments a match, e_{xy}, between a detected pedestrian, p_x, in frame i and a track, t_y, from frame $i - i$ is de-

fined as either; (a) *Correct*, if p_x and t_x represent the same real-world pedestrian; or (b) *Incorrect* if p_x and t_x represent different real-world pedestrians. Results are presented in tables 6.9, 6.10 and 6.11[1]. Evaluation of the results reveal that from the 390 tracks created in all 10 sequences (consisting of 9258 detected regions), 8868 matches between tracks and detected pedestrians were made. From these matches only 41 (or 0.46%) were incorrect. Evaluation of the reasons behind all incorrect matches reveal that, in general, incorrect matches occur when the pedestrian that represents the correct continuation of a track is not present in the scene, but a new un-tracked pedestrian is introduced into the scene in a position that does not invalidate the kinematic constraints. Due to the matching scheme, as long as the appearance models have a 50% match or higher, the track can then be matched to this new pedestrian, leading to an incorrect match.

The *Grafton* 2 sequence incorporates the highest number of incorrect matches. Three such examples can be seen in figure 6.13. The first example occurs in figure 6.13(h) where the track on the far right (surrounded by a green bounding box) is matched to the pedestrian in the red coat, p_1, instead of the pedestrian just behind in the white coat, p_2. The second occurrence is in figure 6.13(j) when the track of p_1 returns to p_2. The third incorrect match is more typical of the sequences as a whole. This occurs in figure 6.13(k) where a highly occluded pedestrian in white exits the scene, at the same time another person enters the scene in close proximity to where the exited pedestrian was positioned in the previous frame. In each of these examples, the incorrect matches occur when the correct continuation of a track is not present in the scene, but a new un-tracked pedestrian is introduced into the scene. It might be considered that this may be avoidable if more sophisticated appearance models are employed. However, as this type of occurrence can happen during high occlusion the application of these type of appearance models may not be able to avoid all incorrect matches.

6.3.3 Full System Overview

Finally, in this section, an overview of the performance of the entire pedestrian detection and tracking system with respect to the 4 test scenarios is presented with the use of the illustrative examples depicted in figures 6.11-6.19. Note that in these figures, the tracks in the plan-view images (i.e. row 2) are depicted as "tails" from the centre of the circle to previous positions in the scene. For the 2D sequences a "tail" indicates the *full* track, but in the 3D Vicon sequences only

[1]If a track represents a region consisting of two under-segmented pedestrians in frame $i-1$ then as long as it is matched to *either* pedestrian in frame i then the match is defined as correct

Sequence	*Total People*	*Total Tracks*	*Total Matches*	*Correct*	*Incorrect*
Grafton 1	577	72	505	499 98.81	6 1.19
Grafton 2	671	60	611	599 98.04	12 1.96
Grafton 3	446	50	396	394 99.49	2 0.51
Grafton Total	1694	182	1512	1492 98.68	20 1.32
Overhead Total	641	41	600	597 99.50	3 0.50
Corridor Total	969	74	895	891 99.55	4 0.45
Total	3304	297	3007	2980 99.10	27 0.90

Table 6.9: 2D sequences biparite matching results.

Sequence	*Total People*	*Total Tracks*	*Total Matches*	*Correct*	*Incorrect*
Vicon 1	198	1	197	197 100.0	0 0.00
Vicon 2	533	3	530	530 100.0	0 0.00
Vicon 4	1301	10	1291	1290 99.92	1 0.08
Vicon 8_A	1980	38	1942	1937 99.74	5 0.26
Vicon 8_B	1942	41	1901	1893 99.58	8 0.42
Total	5954	93	5861	5847 99.76	14 0.24

Table 6.10: 3D sequences biparite matching results.

Sequence	*Total People*	*Total Tracks*	*Total Matches*	*Correct*	*Incorrect*
2d Total	3304	297	3007	2980 99.1	27 0.90
3d Total	5954	93	5861	5847 99.76	14 0.24
Total	9258	390	8868	8827 99.54	41 0.46

Table 6.11: Biparite matching results for all sequences. Two figures, A B, are provided for columns 5 and 6 in tables 6.9, 6.10 and 6.11. A represents the total number of matches classified as either correct or incorrect, B represents the percentage of A with respect to the total number of matches for that sequence. For example, in table 6.9 for row 2 column 5, 499 matches were classified as correct, which equates to 98.81% of the 505 total number of matches in the *Grafton* 1 sequence.

the last 10 positions are presented in the "tail" of a pedestrian for ease of visualisation as some tracks are maintained for over 200 frames.

It is clear that the proposed system is robust to two pedestrians in close proximity. In the sequences people tend to get into close proximity regardless of the pedestrians density flow. Some of these occurrences were planned to test the system, for example in the *Overhead* sequence of figure 6.11. Others occurred naturally, for example in figure 6.12(b), in order for pedestrians to continue on their chosen paths. Yet others were caused by high pedestrian density flow – see figures 6.13, 6.18 and 6.19. In these sequences it can be seen that tracks are generally coherent and are not lost even on close interaction.

Figures 6.13, 6.18 and 6.19 illustrate the robustness of the detection and tracking techniques when subjected to unconstrained crowded conditions. They all depict multiple pedestrians travelling in various directions being tracked robustly. In figure 6.13 up to 13 people are successfully tracked concurrently. For example, in figures 6.13(e)-(h) a pedestrian wearing a black suit (surrounded by a yellow bounding box) makes a u-turn in the sequence and is successfully tracked. A similar occurrence is depicted in figure 6.19 of the *Vicon* 8_B sequence, where a person (surrounded by a grey bounding box) swiftly walks around right of the Vicon elliptical area, and in figure 6.19(g) makes a quick u-turn at the bottom of the scene and walks inward towards the centre of the elliptical area. Again in this sequence the pedestrian is successfully tracked.

The overall precision of the system was 97.82% – with just 8.97% of the "non-pedestrian regions" caused by pedestrian over-segmentation. This illustrates strong and robust systems performance for a variety of pedestrian poses and orientations, regardless of the camera placement and orientation. However, over-segmentation of pedestrians can still occur. One such over-segmentation track can be seen (surrounded by a green bounding bx) on the left hand side of figures 6.14(f)-(h). A second can be seen on the right hand side (surrounded by a white bounding box) of figures 6.14(m)-(n).

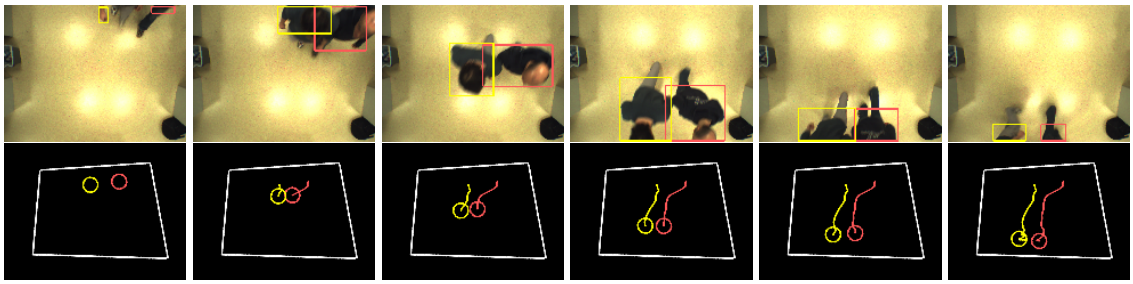
The robustness of the system to changing illumination conditions is demonstrated in figure 6.13. In this sequence the lighting conditions change dramatically. Some of the pedestrian tracks are lost, due to a lack of foreground gradient texture within the dark cast shadow regions. However, it should be noted that the remainder of the pedestrians are tracked correctly, and that *no* regions caused by background areas are detected as valid pedestrians within this period of the sequence – there is background region in the whole sequence caused by the reflection of a pedestrian off a window. This is a very impressive result considering the background update facility was effectively

turned off for the *Grafton* 3 sequence – see section 4.3.6.1. As described in section 6.3.2, two such background tracks can be seen to occur on the foreground pillar in figures 6.12(a) and 6.12(d).

Finally, the robustness of the system with respect to occlusions is depicted in the *Vicon* sequences of figures 6.15-6.19. Figure 6.15, from the *Vicon* 2 sequence, illustrates both partial occlusion and full occlusion. In this sequence, the system correctly detects a partial occlusion in figure 6.15(b), and a full occlusion in figure 6.15(d). Due to these detections the system is able to continue the pedestrians track throughout the full occlusion (which only lasts for a single frame). A more complicated example, with multiple occlusions, is depicted in figure 6.16 from the *Vicon* 4 sequence. Again, the system is able to correctly determine full occlusions and to continue tracks successfully throughout occlusion, even when multiple occlusions – see figure 6.16(j) – are simultaneously occurring. A feature inbuilt into the system is to terminate tracks if they become occluded for more than three frames. Such an occurrence from the *Vicon* 4 sequence is depicted in figure 6.17 where two tracks are terminated and two new ones are successfully created. This is a design choice of the system. In the *Vicon* scenarios, as it is known how many people are in the sequence, this feature could simply be extended to allow tracks to be extrapolated through longer occlusions. However, in the real-world sequences, for example in figure 6.13, there is no guarantee that; (a) the person will ever emerge from occlusion; (b) the person can be extrapolated correctly for longer periods of time when occluded; and (c) that if someone did emerge from occlusion that they represent the same person that was originally occluded. A set of examples from the *Vicon* 8_A and *Vicon* 8_B sequences are provided in figures 6.18 and 6.19 respectively. In each of these sequences the robustness of the system to occlusions is further demonstrated. Finally, further illustrative examples of the final pedestrian detection and tracking system from other challenging scenarios are presented in figures D.1-D.5 of appendix D. In addition, the results of the final pedestrian detection and tracking system from the 10 test sequences are available on-line at *www.eeng.dcu.ie/~kellyp/thesis*.

6.4 Summary

In this chapter a *continuous detect-and-track* framework was proposed in which pedestrians are temporally tracked using both appearance and positional information. A *Maximum Weighted Maximum Cardinality Matching* scheme was employed, with additional kinematic constraints, to obtain the best match from previous tracks to currently detected pedestrians. This technique also

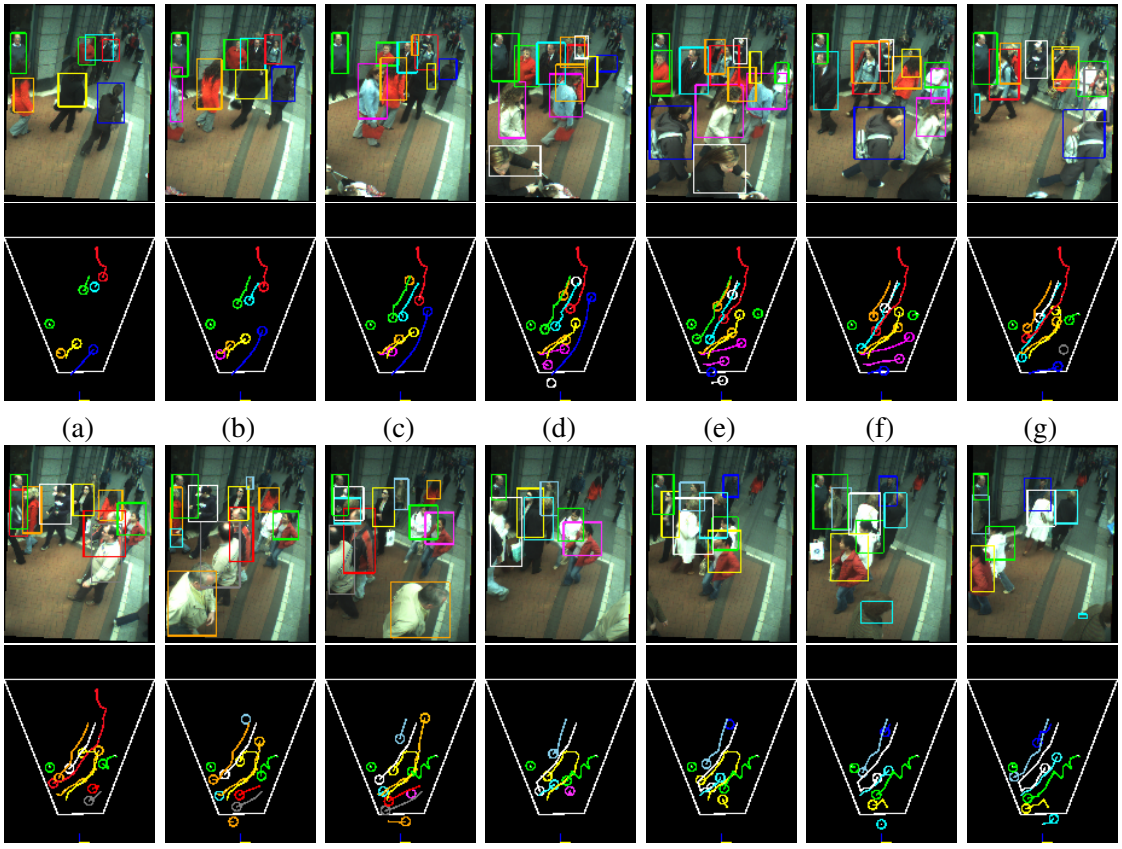


(a) (b) (c) (d) (e) (f)

Figure 6.11: *Overhead* sequence. Every fourth frame between (a) 341 - (f) 361.

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

Figure 6.12: *Corridor* sequence. Even frame numbers between (a) 292 - (f) 306.



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

(h) (i) (j) (k) (l) (m) (n)

Figure 6.13: *Grafton* sequence 2. Even frames numbers between (a) 18 - (n) 44.

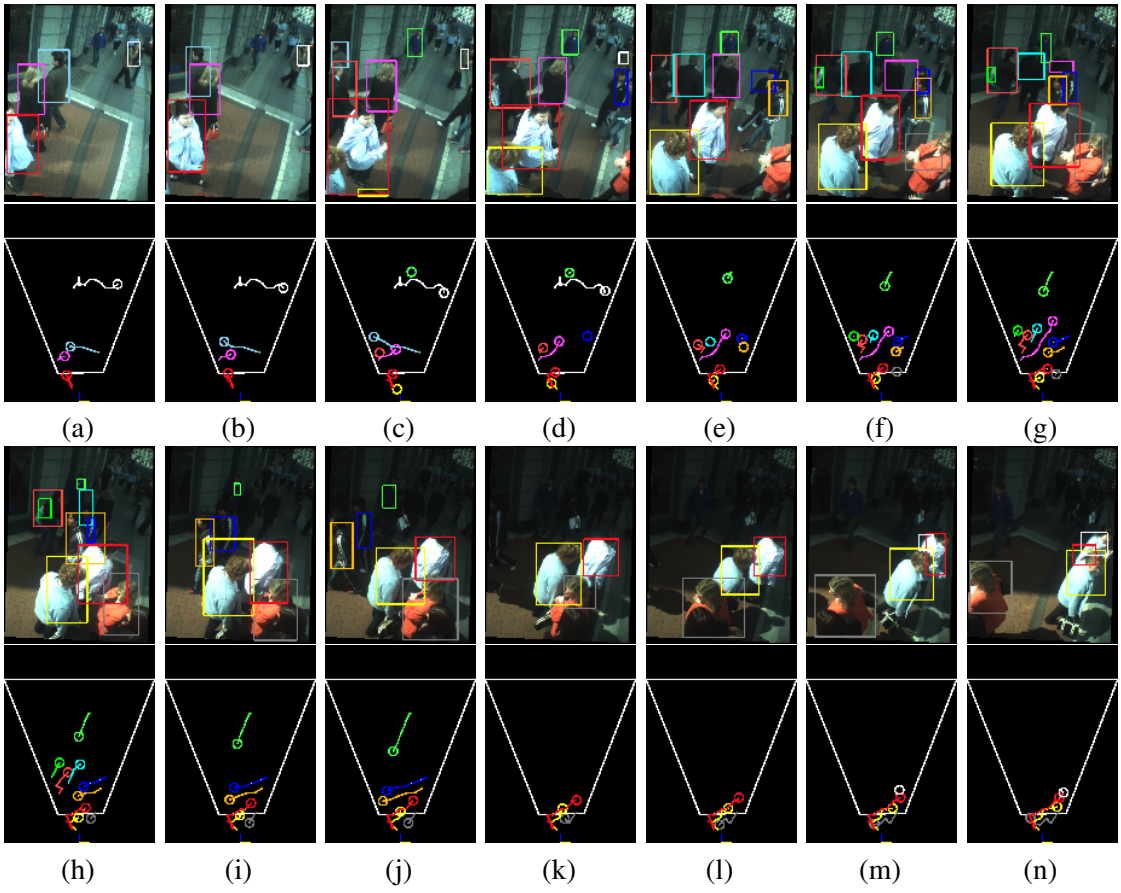


Figure 6.14: *Grafton* sequence 3. Frames numbers (a) 60 - (n) 73.

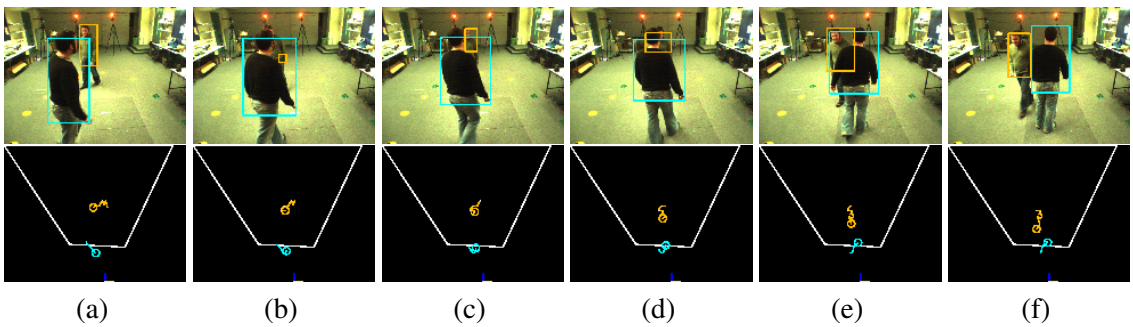


Figure 6.15: *Vicon* 2 sequence. Even frame numbers between (a) 92 - (f) 102.

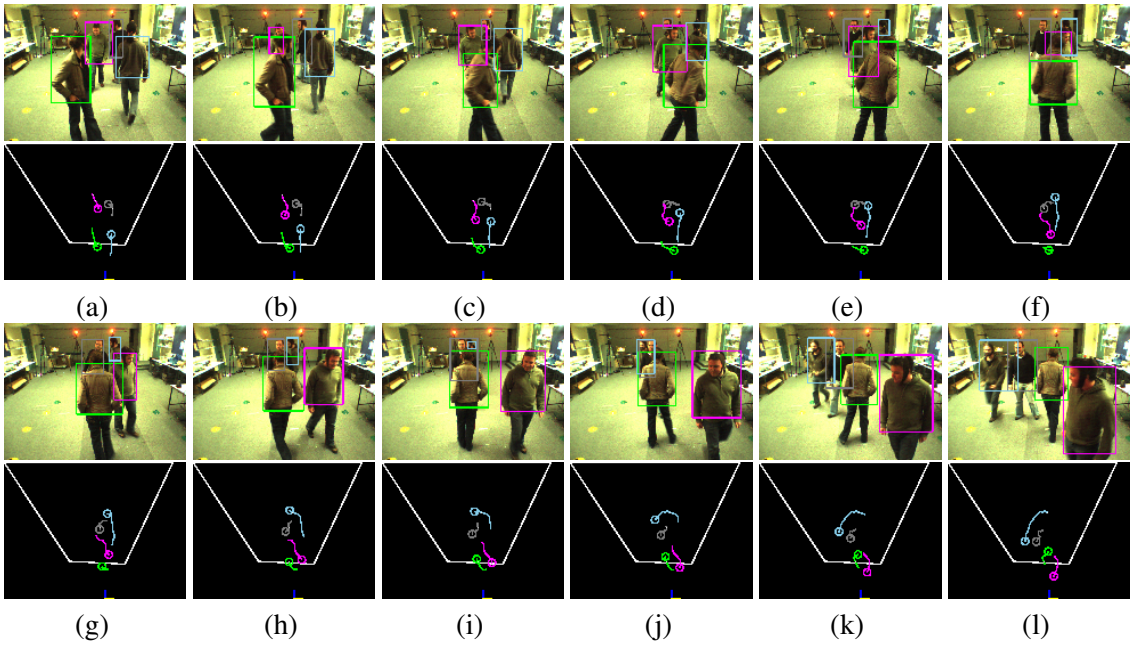


Figure 6.16: *Vicon* 4 sequence. Even frame numbers between (a) 180 - (l) 202.

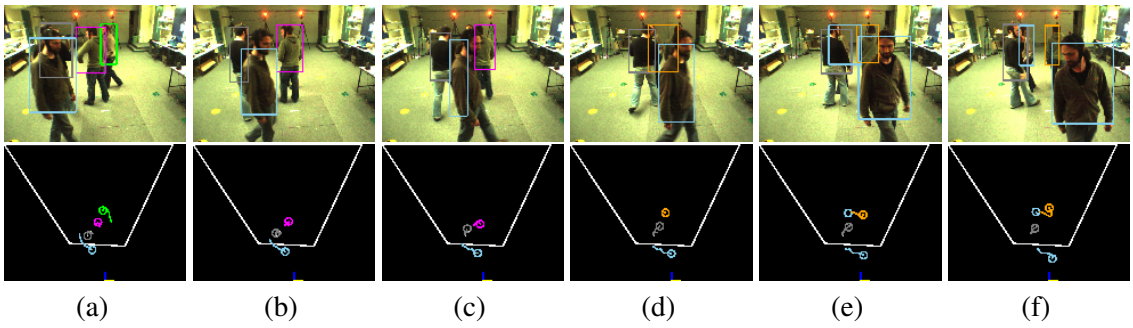


Figure 6.17: *Vicon* 4 sequence. Odd frame numbers between (a) 295 - (f) 305.

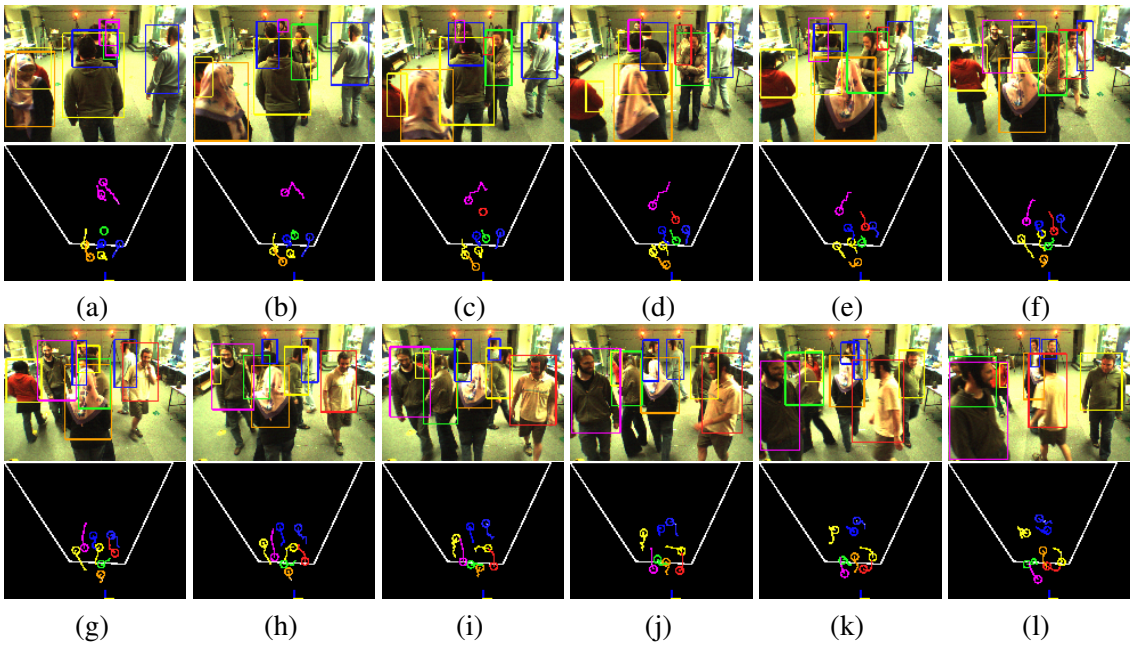


Figure 6.18: *Vicon* 8_A sequence. Every third frame between (a) 95 - (l) 128.

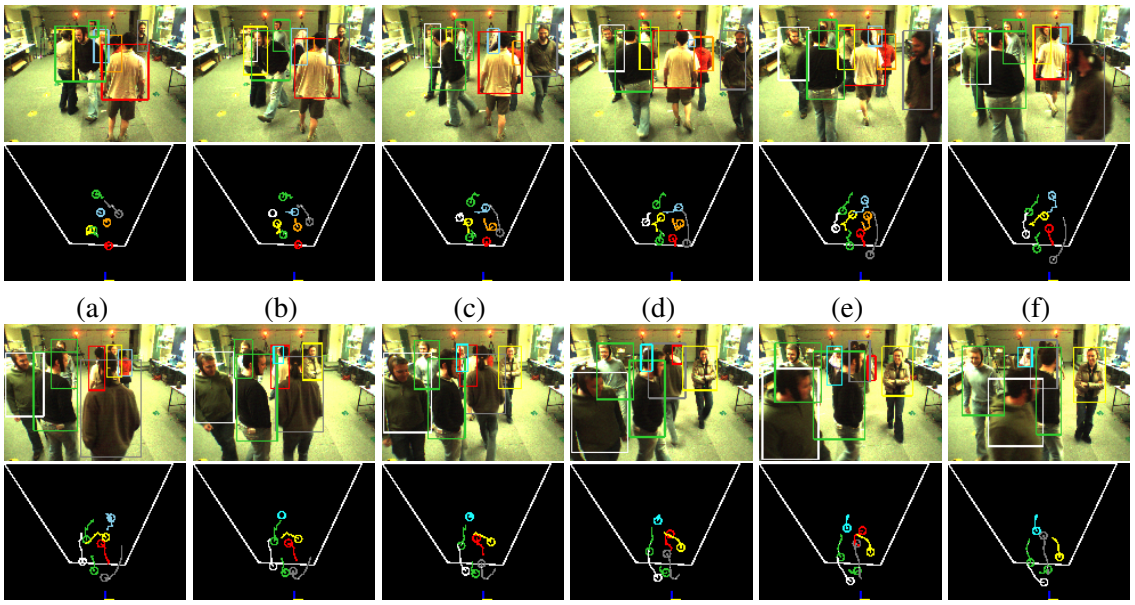


Figure 6.19: *Vicon* 8_B sequence. Even frame numbers between (a) 114 - (l) 136.

employed a variety of pedestrian detection rollback mechanisms and global analysis techniques, such as explicit occlusion analysis, to further reduce over-/under-segmentation of detected pedestrians and increase tracking robustness. This technique was shown to be robust to a variety of camera heights, rotations and orientations and also to varying pedestrian flow and illumination conditions. The technique was evaluated using the same groundtruth methodologies as in chapter 5. The results of this analysis indicated an average increase of 2.93% and 2.35% in precision and recall values. This therefore results in final precision and recall values of 97.82% and 91.49% for the 10 challenging sequences. In addition, the system tracking methodology was evaluated using the same 10 sequences. Overall, the results of the final pedestrian detection and tracking system indicate extremely robust and accurate performance of the proposed approach in a variety of scenarios. In the final chapter of this thesis, a summary of the proposed approach and directions for future work in this area are outlined.

CHAPTER 7

Conclusions and Future Work

In this thesis, a system that has been designed to robustly detect and track pedestrians in unconstrained environments using a single short-baseline stereo camera is proposed. During the early stages of this work, see section 1.1, it was stated that the system should attempt to minimise assumptions about environmental conditions, pedestrian appearance, pedestrian flow density, the pedestrian and background colour intensity information, the length of time a person exists within the scene, the number of persons within the scene, or how a pedestrian enters the scene. In addition, it was stated that the system requires *no* external training[1] and yet is robustly able to handle:

1. occlusion, even when multiple people enter the scene in a crowd;
2. lack of variability in colour intensity between pedestrians and background;
3. rapidly changing and unconstrained illumination conditions;
4. pedestrians appearing for only a small number of frames;
5. relatively unconstrained pedestrian movement;
6. relatively unconstrained pedestrian pose, appearance and position with respect to the camera;
7. varying camera heights, rotations and orientations;
8. static pedestrians.

[1] However, it is acknowledged that the designer has brought in his own area of expertise into setting a number of hard-coded thresholds throughout the system framework.

After a rigorous evaluation, it has been shown that, in general, the system meets these requirements. The final system obtains pedestrian precision and recall values of 97.82% and 91.49% respectively from 10 challenging sequences. This indicates accurate performance of the proposed approach under a variety of scenarios.

7.1 System Assumptions, Limitations and Potential Issues

Although the proposed system has been shown to be robust in a number of scenarios, in its current state, the system does contain a number of assumptions and possible limitations within each module. In the following section the inherent assumptions within the proposed system framework are highlighted and discussed. Following this, each module is examined independently with a view to highlighting possible limitations and issues inherent within the proposed pedestrian detection and tracking system. It should be noted that an area of future work envisioned by the author includes the investigation of techniques to address these assumptions and drawbacks.

7.1.1 Assumptions Overview

Although the proposed system was designed to minimise constraining assumptions, a small number of inherent assumptions still exist within the system framework. They include;

1. that pedestrians in the scene are standing upright with respect to the groundplane;
2. that all moving objects in the scene (within the volume of interest) are caused by foreground pedestrians;
3. that pedestrians in the scene are moving at a velocity of less than 3 metres per second.

In addition to this, there are a number of limitations on the type of scenario it can be used in. These include; (a) that a relatively flat groundplane is present within the scene, where no object of interest is located below this groundplane; (b) the camera must be orientated so that the groundplane is visible in the image plane; and (c) the system is only able to reliably detect pedestrians for a short-medium range, up to a maximum distance of 8 metres from the camera.

7.1.2 Disparity Estimation Module

With regard to the limitations of the system, due to the use of groundplane space within the proposed disparity estimation technique – see section 4.3.2 – corresponding pixels are not searched

below the disparity of the scene's groundplane. This technique can be problematic for some scenarios, for example in scenes where pedestrians can descend a staircase to below the groundplane level. If such a scenario occurs, then the correct disparity of those pedestrians will not be obtained via the proposed technique.

A second potential issue lies in the strong inter-scanline consistency enforced by the vertical smoothness cost in the dynamic programming algorithm. As mentioned in section 4.4.3.2, the strength of this smoothness cost can result in a decrease in the accuracy of the final disparity map. Two such examples were illustrated in figures 4.29 row 5 and 4.27 row 1 – for further details see section 4.4.3.2.

Finally, a third area of potential difficulty lies within the dynamic disparity limit constraint of the proposed technique. For a given scanline, this disparity limit is determined as the maximum of the previous scanline's *dense* foreground disparity, the current and next scanlines GCP disparity, and the maximum disparity within the corresponding three scanlines in the background model – see section 4.3.7.1. Consequently, for the proposed dynamic programming algorithm to assign the correct disparities to a foreground object, it requires that a GCP of equal or greater disparity must exist in the current (or previous) scanlines. Therefore, the success of the proposed dense disparity estimation approach can be seen to inherently linked to that of the GCP identification process.

7.1.3 Pedestrian Detection Module

The use of the biometric person model – as defined in section 5.3.1 – has a number of advantages over other pedestrian models applied in the literature. However, an inherent assumption for this model is that all pedestrians within the scene are standing with respect to the scene's groundplane. This assumption on pose is violated for pedestrians who are sitting down, for example in wheelchairs, or office chairs. For these scenarios, due to a decrease in pedestrian height above the groundplane, the proposed system will mis-interpret these persons to be shorter pedestrians, such as children. As such the biometric person model is scaled to a smaller size than required, which could in turn lead to over-segmentation of these pedestrians. Conversely, if a person is located with their feet above the scene's groundplane, for example if a person is ascending stairs that rise above the groundplane or if a person is jumping, then the system will mis-interpret these pedestrians as being taller. As such the scale of the resultant biometric person model is increased and consequently could lead to under-segmentation of these pedestrians.

A second potential limitation lies within the choice of background models. As shown in the ex-

perimental scenarios, the background model is robust to background illumination changes. However, as each pixel in the background gradient model is based on a uni-modal Gaussian distribution, the model lacks robustness when faced with multi-modal backgrounds where the means of the distributions differ greatly. This type of background distribution can occur, for example, with leaves moving on a tree in the background. In addition to this issue, due to the slow update parameter built into the two-layered background model, the technique can also be affected by *ghosts* (see section 2.2.2).

A third potential issue of the proposed module can be seen in the fact that the system assumes that all moving objects within the volume of interest of the scene are caused by foreground pedestrians. As outlined in section 5.3, due to the range of the camera, and the envisioned applications, this is not an unreasonable assumption in crowded situations if the likelihood of all objects in the scene being pedestrian is high, e.g. in pedestrianised urban areas. However, if other objects, such as cars appear within the volume of interest of the scene, then the proposed technique will attempt to segment those objects into pedestrian objects. This will therefore result in a decrease in the precision of the proposed system.

Finally, the system's volume of interest – see section 5.2.1 – is constrained to a maximum of 8 metres from the camera. The reason for this maximum distance is constrained by the Digiclops camera stereo rig parameters and a disparity estimation technique that does not obtain sub-pixel accuracy. This maximum distance can become an issue if the required surveillance area is located further from the camera than the maximum 8 metres. A possible solution to this issue would be to alter the camera parameters within the camera stereo rig. For example, by increasing the camera's baseline or focal length by a factor of 2, 10, or 150 then the maximum distance for the system would be increased to 11.5, 26 and 100 metres respectively. However, this increase in either of these parameters would result in a decrease in the overlap between stereo image pairs and as such the disparity of pedestrians closer to the camera may become unobtainable. If such an event occurs, then those pedestrians would not be detected via the proposed approach. A second solution to this issue would include a post-processing stage within the disparity estimation technique that obtains sub-pixel accuracy within the resultant disparity maps.

7.1.4 Pedestrian Tracking Module

The tracking framework caters for pedestrians that are either standing, accelerating or walking (at a maximum speed of 3 metres per second). However, no provision is made for pedestrians moving

at a higher velocity. As such pedestrians running through a scene may not be tracked robustly.

In addition to this potential issue, the proposed tracking technique can be limited by the fact that an occluded pedestrian's track is only extrapolated for a maximum of 2 frames (after which the track is removed from the system). For some applications, it may be critical that pedestrian tracks are robustly held through longer occlusions, for example while tracking potential security threats within public spaces, such as airports. For such scenarios, a more sophisticated extrapolation or multi-camera approach may be required in order ensure more robust tracking.

With regard to the pedestrian appearance models, the choice of the appearance model in this work is not robust to changes in scale when a person is occluded. As outlined in section 6.2.4.3, the colour model is simply updated by replacing all relevant foreground pixels in the track's HSV colour model with those within the height bounds of the matched pedestrian in the current frame. However, if an area of a tracked pedestrian becomes occluded while the person is walking towards/away from the camera, then when the person becomes un-occluded the normalisation of the colour model will be somewhat incorrect. This is due to differences in scales from when the occluded part of the body was last updated. Therefore, the area that was occluded will obtain a higher or lower weighting depending on whether the pedestrian was walking away from or towards the camera respectively.

Finally, within the evaluation test sequences, a small number of erroneous tracking matches occurred. Within the test datasets, these matches generally involved newly entered pedestrians, or pedestrian tracks that have not been well established - i.e. had existed for less than three frames. One such example of this type of event within the evaluation test sequences occurred when one pedestrian exited the scene and a second, of similar appearance, simultaneously entered the scene at a similar location. Due to this occurrence, the old track was erroneously matched to the newly entered pedestrian. A second scenario occurred when two newly created tracks (of similar appearance) swapped positions. This erroneous match occurred due their previous positional and vector information indicating the incorrect match to be more likely (but without violating the proposed kinematic constraints).

7.2 Thesis Overview and Research Contributions

In this thesis, a review of the application of both 2D and 3D image processing techniques for pedestrian detection and tracking is provided in chapters 3 and 4 respectively. In addition, chapter

3 describes multiple view geometry. A review of stereo correspondence techniques is provided in chapter 4.

The proposed pedestrian detection and tracking system consists of three main modules. Each module in the system can be viewed as a major research contribution; however in addition, each module consists of a number of additional contributions. The majority of chapter 4 concerns the first system module. In this chapter, a dynamic programming based stereo correspondence technique is defined that has been specifically developed for applications involving pedestrian detection. The technique reduces artifacts in the calculated disparity map via a number of novel enhancements to the dense disparity estimation algorithm. These include (a) reducing the disparity search space via a 2D projective transformation of the input rectified images into *groundplane space*; (b) a novel technique to obtain highly reliable disparity matches, known as Ground Control Points (GCPs), to help guide final disparity estimation results; (c) the use of a *dynamic* disparity limit constraint in the disparity estimation process; and (d) a novel scanline cost function for the dynamic programming algorithm that enforces inter-scanline consistency in the final disparity map. The evaluation of the proposed disparity estimation technique, both qualitatively and quantitatively, was shown to outperform a variety of other disparity estimation algorithms for the given test set. This test set included a variety of challenging synthetic scenarios designed to mimic typical pedestrianised scenarios and their associated difficulties that included; varying lighting conditions; shadows; a lack of texture at depth discontinuities; homogeneous foreground and background regions; and most importantly pedestrians exhibiting a variety clothing; orientations; distances; and scales.

In chapter 5, the disparity map is post-processed and used as an input to the pedestrian detection module, which is the second major contribution of this work. In this approach, the post-processed disparities are clustered together into pedestrian regions via a novel iterative region growing framework that incorporates 3D information, groundplane estimation, non-quantised plan-view statistics and a human biometric model that is automatically tailored for each pedestrian during their segmentation. In addition to these contributions, a background subtraction framework is outlined, which is highly robust to changing illumination conditions. This technique is employed in the system to remove background regions and noise. This process is evaluated in 4 different scenarios using both 2D evaluation techniques and a novel 3D methodology.

The final module of the system is that of pedestrian tracking using a continuous detect-and-track methodology using a weighted bipartite graph. This technique is outlined in chapter 6.

Contributions in this chapter include; (a) a maximum-weighted-maximum-cardinality matching scheme that incorporates a novel weighting scheme for matching pedestrians to previous tracks; (b) a series of kinematic constraints that model possible pedestrian movement through the scene and as such is used to remove implausible matches; and (c) a post-processing stage to increase track robustness to occlusion and under-segmentation. In this chapter, the robustness and accuracy of the final system is evaluated using both 2D and 3D evaluation techniques.

7.3 Future Work

During the development and evaluation of the pedestrian detection and tracking system, numerous avenues of potential investigation for future research were uncovered. Some of these directions may lead to further improvement of the proposed techniques – some of these areas have already been discussed in section 7.1. Other directions can be seen as the application of the proposed system as an integral element in a variety of application areas. In this section, a number of these future directions are described in further detail.

7.3.1 Algorithmic Improvements

The evaluation of the proposed disparity estimation technique in this work indicates its robustness when compared to a variety other disparity estimation algorithms. However, it is noted that this evaluation was made from a single static binocular camera system. Further evaluation of the system should be carried out using a variety of stereo camera rigs exhibiting a range of baselines, focal lengths and other camera parameters. In addition, the possibility of further improving results by incorporating a third camera should be investigated. This additional camera would allow a geometric constraint, known as the *trinocular constraint* (see section 3.2.3), to be employed to reduce correspondence matching ambiguities. Research in this area could be applied to determine what, if any, increase in accuracy can be achieved using a trinocular system and at what computational expense. In addition, if any improvements in overall disparity accuracy are achieved, it should be investigated whether this would result in similar increases in the overall system precision and recall.

As outlined in section 5.3, many pedestrian detection techniques, including the proposed system, make the assumption that all objects, which are not due to background objects, are caused by pedestrians. For the proposed system, and its envisioned applications, this is not an unreasonable

assumption. In crowded situations the likelihood of all objects in the scene being pedestrian is high, e.g. in pedestrianised urban areas. A variety of techniques, such as Support Vector Machines (SVM), can be used to classify a region as a pedestrian or not. However, these are generally difficult to train, especially when robustness to camera position, orientation and high pedestrian occlusion is required. Therefore, using these techniques to classify pedestrians may result in a large drop in recall. However, an interesting direction of research may be to investigate the use of the proposed pedestrian segmentation and tracking techniques within the framework of such classification methodologies. For example, a region may not exhibit pedestrian traits in single frames, possibly due to occlusion, but could be correctly classified as a pedestrian by observing its movement and behaviour over time. Alternatively, the framework could be used as a basis to classify objects that are *not* pedestrians, thereby creating a framework whereby all objects within the scene are hypothesised to be pedestrian unless it is to be shown to be otherwise.

In the proposed system, a predefined volume of interest (VOI) is employed – see section 5.2.1 – whereupon all 3D points outside of this VOI are considered irrelevant in the search for pedestrians. This VOI is defined with respect to two planes; the camera plane and the 3D groundplane. In our experiments, all 3D points that are below 0.9 meters ($\simeq$ 3 foot) in height above the 3D groundplane are removed in a post-processing step. This stage tends to correctly eliminate regions belonging to the background, such as groundplane points that include shadows cast by pedestrians and other objects, and other non-pedestrian objects, which include push-prams and buggies. However, applying this post-processing technique can also remove small children or people in wheelchairs if they are below the 0.9 metre threshold. Therefore, future work should evaluate this threshold to determine the ideal threshold for a given scenario, to increase recall without an adverse effect on precision. In addition, a final post-processing stage (in possibly both pedestrian detection and tracking modules) should be investigated whereby the VOI can be altered to uncover lost pedestrians that are under this 0.9 metre threshold. Finally in this area, as outlined in section 3.3, robust techniques for automatically calibrating the camera with respect to the groundplane, perhaps by adopting the v-disparity approach or by applying some feature matching techniques, should be investigated.

Future work in pedestrian tracking should investigate the possibility of fusing more than one type of technique. As outlined in section 5.2.1, the proposed technique does not detect pedestrians at a distance greater than 8 metres from the camera. As such, the *continuous detect-and-track* framework is unable to track pedestrians outside this distance. However, the pedestrian detection

and tracking technique can be seen as highly accurate within these bounds, especially at distances of less than 6 metres – as seen in the results of the *Vicon* sequences where 99.65% precision and 95.02% recall was obtained. By maintaining more sophisticated pedestrian appearance models, such as those adopted in [79, 87], it would be possible to switch, or augment, the proposed tracking framework with a more independent tracking scheme based on *single detect-and-track* methodologies when a pedestrian reaches a particular distance from the camera. Using this technique, a track may be maintained more robustly at greater distances from the camera. However, for this technique, alternative updating techniques for appearance model maintenance should also be investigated.

Finally, techniques to reduce the computational complexity of the proposed algorithms should be investigated. In our experiments, the proposed system was implemented in un-optimised C++, designed in a highly object-oriented framework, and run on a 2GHz laptop. In general, the overall processing time for each frame varies – the more pedestrians within the frame, the more FARs to search for GCPs, and the more foreground disparity points need to be clustered. This leads to longer processing times. On average the processing of a single 640×480 pixel frame takes between 10–20 seconds. Obviously this is far from real-time processing. However, throughout the system development, the algorithmic design took precedence over complexity, which was rarely addressed. Apart from optimising code, a number of research paths exist that would maintain the main algorithmic features, but decrease complexity.

The main bottleneck in the system is the pedestrian detection module (taking on average between 6–16 seconds). However, a number of techniques can be used to decrease the complexity of this module. Currently in the iterative clustering framework, an orthographic projection path onto the groundplane for each 3D point in each region is obtained. Each of these paths are traversed, and any two regions can be tested and possibly merged if they both appear on that path. This is obviously computationally inefficient as the same region can be tested, and rejected, multiple times. A more efficient technique would be to test each region to all other regions, within a range defined by the biometric human model, only once per iteration of the algorithm. Initial tests on this technique alone indicate a possible reduction of up to 50% in processing time. A second investigation in this area could be based on the idea of reducing the clustering space from 3D to 2D. The most logical way to achieve this may be to employ the use of further plan-view statistics, in particular the use of occupancy and height maps, within the iterative clustering framework. Using this approach, the height map would be applied to control the scaling of the biometric human

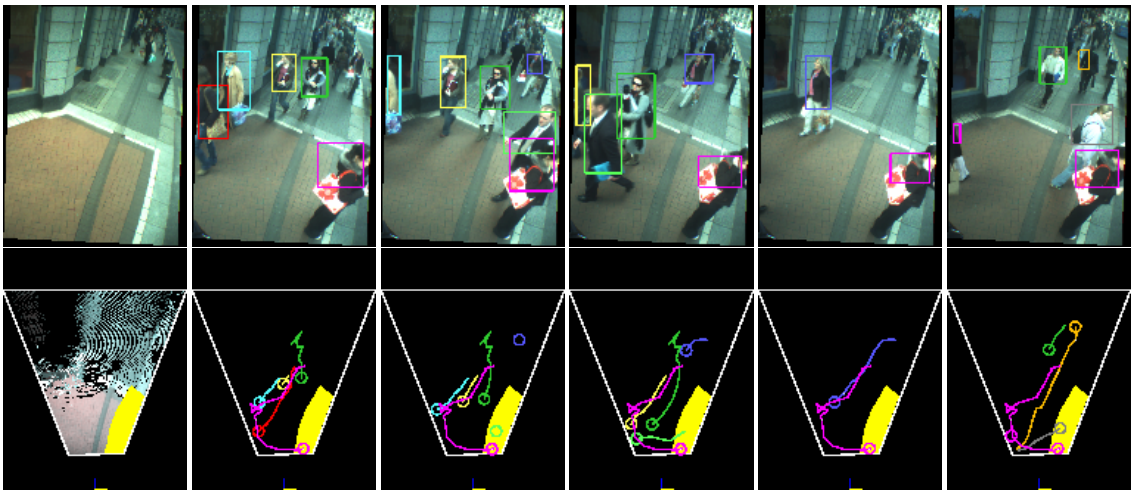
model, while the occupancy map can be used to correctly control the positioning of the model. The possible advantage of this technique is that the number of points to be clustered may be reduced, due to the quantisation of the height and occupancy maps. However, some loss of detail will occur, due to the quantisation of the 3D points into a discrete 2D bins. This loss of accuracy could be outweighed, however, by improvements in computational efficiency. Further reductions in computational efficiency may also be found in the use of a hierarchical coarse-to-fine framework, whereby the clustering is first implemented at a lower resolution. These results could then be used as a basis for clustering at higher image resolutions.

7.3.2 Application Based Event Detection

In addition to future research directions with regard to the pedestrian detection and tracking algorithms, the use of the system as a key enabling technology for a suite of applications is envisaged. Using this technology potentially allows other layers in an application's framework to infer beliefs about people in the scene and what their actions have been.

The detection of application specific events promises many benefits for both single individuals and larger groups of people in a variety of application scenarios. However, depending upon the end application a variety of event detectors may need to be defined. For some applications these event detectors can be hard-coded into the application framework, for example surveillance applications that determine pedestrian flow densities during specific time periods. However for many applications the exact event detector *cannot* be hard-coded into the system as; (1) the event definition is dependent on an undefined scene; or (2) the event is itself undefined.

The first scenario is typical of many Ambient Intelligence (AmI) applications, such as the automated pedestrian traffic light system mentioned in section 1.1, as although the event required to be detected is known (i.e. detect pedestrians in a designated area waiting to cross the road), information about the scene (i.e. the exact designated area) is unknown. The second scenario is typical of general purpose surveillance applications where events are determined either at run-time or *after* the event has occurred. Run-time events include adding "forbidden" areas in a scene or flagging "unusual" events such as lingering pedestrians. However, for some applications the event itself is undefined. Take for example a typical CCTV surveillance system, if the surveillance video is augmented with robust pedestrian tracking and statistical information it becomes possible to quickly search the video for specific events via the augmented meta-data, such as the lost/abducted child or vandalism scenarios conjectured in section 1.1. In this way, the pedestrian detection and



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

Figure 7.1: Pedestrian crossing application; (a) Hotspot; (b)-(f) Pedestrian waiting to cross the road.

tracking system could be used as the basis for content-based multimedia storage and retrieval applications.

To this end, a pedestrian *indexing* scheme and suite of tools for detecting events and retrieving data from a given scenario is envisaged. Using these tools, events can be detected at run-time for use in Ambient Intelligence (AmI) applications, or be applied as a framework in which to search for events *after* the event has occurred for content-based retrieval applications. One such tool is the creation of 3D *hotspot* regions from 2D plan-view images – see the 2D yellow coloured area in figure 7.1 (a). In this figure, a background colour model is projected onto the image-plane to allow the gauging of distance and orientation in the plan-view image. An example application of 3D hotspot can be seen in figures 7.1 (b)-(d) where it is applied in the framework of the automated pedestrian traffic light system. The hotspot is used to determine if a pedestrian is within an area where they are deemed to be *possibly* waiting to cross the road. A detected event that in turn leads to a changing of the traffic lights could then be defined as when a person walks onto the hotspot, stops and waits for a period of time, such as the pedestrian in the bottom right of figure 7.1 (b)-(d). In the proposed system, a hotspot is simply created by circling a region of interest within the plan-view image. This system therefore allows an application, such as the pedestrian crossing system, to be easily tailored for multiple scenes.

Using these hotspot regions and an event (or query) syntax, a suite of event detectors can be parsed and ran during run-time. An example of the syntax could take the following form; $Event(cross) = hpt(1), p(g5), t(10s)$ – which declares an event called *cross* to be detected if on

hotspot number 1 ($hpt(1)$) if there are greater than 5 pedestrians ($p(g5)$) who have been waiting for more than 10 seconds $t(10s)$. Of course an event can consist of any number of features, including; hotspots, pedestrians (including pedestrian statistics such as colour, height, velocity and position), time of day, timings (such as the length of time an object is in the scene), and pedestrian interactions (such as pedestrians walking in a group or on their own). This potentially allows a powerful user-defined event detection system that can be simply tailored for a suite of pedestrian detection and tracking applications.

APPENDIX A

Groundplane Homography Estimation

The 3×3 homography matrix, H, has 9 entries, but it has only 8 degrees of freedom as it is a homogeneous matrix and therefore defined only up to scale. Each 2D homogeneous image point, $u = (x, y, 1)^T$, has two degrees of freedom corresponding to the x and y components. A point correspondence $u_1 \rightarrow u_2$ between the two projective planes gives two constraints on H, since for each point u_1 on one projective plane, the two degrees of freedom of the second point must correspond to the mapped point $Hu_1 \cong u_2$. Therefore at least four point correspondences are needed to constrain H fully. A solution to H can then be obtained using 4 corresponding points and the Normalised DLT algorithm [148].

However, if I_1 and I_2 are rectified then corresponding points have the same value of y. The result of this rectification process is that the homography can be broken down into a shear and translation of one image to the second image across a single image axis (usually the x-axis). Therefore, the resultant homography takes the form;

$$\begin{vmatrix} \alpha & -\beta & \gamma \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} \tag{A.1}$$

From the matrix A.1 it can be seen that if a point in I_1, $u_1^n = (x_1^n, y_1^n, 1)$, represents a point on the groundplane, then the corresponding groundplane point in I_2, $u_2^n = (x_2^n, y_1^n, 1)$, where x_2^n can be calculated as

$$x_2^n = \alpha x_1^n + \beta y_1^n + \gamma \tag{A.2}$$

The values for α, β and γ can be obtained using 3 matching points. From Equation A.2:

$$x_2^1 = \alpha x_1^1 + \beta y_1^1 + \gamma \tag{A.3}$$

$$x_2^2 = \alpha x_1^2 + \beta y_1^2 + \gamma \tag{A.4}$$

$$x_2^3 = \alpha x_1^3 + \beta y_1^3 + \gamma \tag{A.5}$$

From Equation A.3,

$$\beta = \frac{x_2^1 - \alpha x_1^1 - \gamma}{y_1^1} \tag{A.6}$$

Using both Equations A.4 and A.6, γ can be obtained in terms of α;

$$\gamma = \frac{x_2^2 y_1^1 - x_2^1 y_1^2 + \alpha x_1^1 y_1^2 - \alpha x_1^2 y_1^1}{y_1^1 - y_1^2} \tag{A.7}$$

Using Equations A.6 and A.7, β can be obtained in terms of α;

$$\beta = \frac{x_2^1 - \alpha x_1^1 - x_2^2 + \alpha x_1^2}{y_1^1 - y_1^2} \tag{A.8}$$

Finally, using Equations A.5, A.7 and A.8

$$\alpha = \frac{x_2^1 y_1^2 - x_2^1 y_1^3 + x_2^2 y_1^3 - x_2^2 y_1^1 + x_2^3 y_1^1 - x_2^3 y_1^2}{x_1^1 y_1^2 - x_1^1 y_1^3 + x_1^2 y_1^3 - x_1^2 y_1^1 + x_1^3 y_1^1 - x_1^3 y_1^2} \tag{A.9}$$

APPENDIX B

Tracking Symbols

An overview of all thresholds and symbols used in chapter 6 are provided in tables B.1 and B.2 respectively.

Threshold	Value	Meaning
td^{i}_{i-1}	–	the time difference between frames $i-1$ and i
$dist_{max}$	200 cm	the *absolute* maximum distance a pedestrian is assumed to walk in a second
$dist_{avge}$	300 cm	the *average* maximum distance a pedestrian is assumed to walk in a second
t_{max}	–	the *absolute* maximum distance a pedestrian can travel between frames $i-1$ and i
t_{avge}	–	the *average* maximum distance a pedestrian can travel between frames $i-1$ and i
t_{noise}	30 cm	the maximum estimated error between a pedestrian's detected and real-world position
θ_{max}	60°	the maximum angle pedestrian can turn in one frame while walking at a high velocity

Table B.1: Pedestrian tracking thresholds overview.

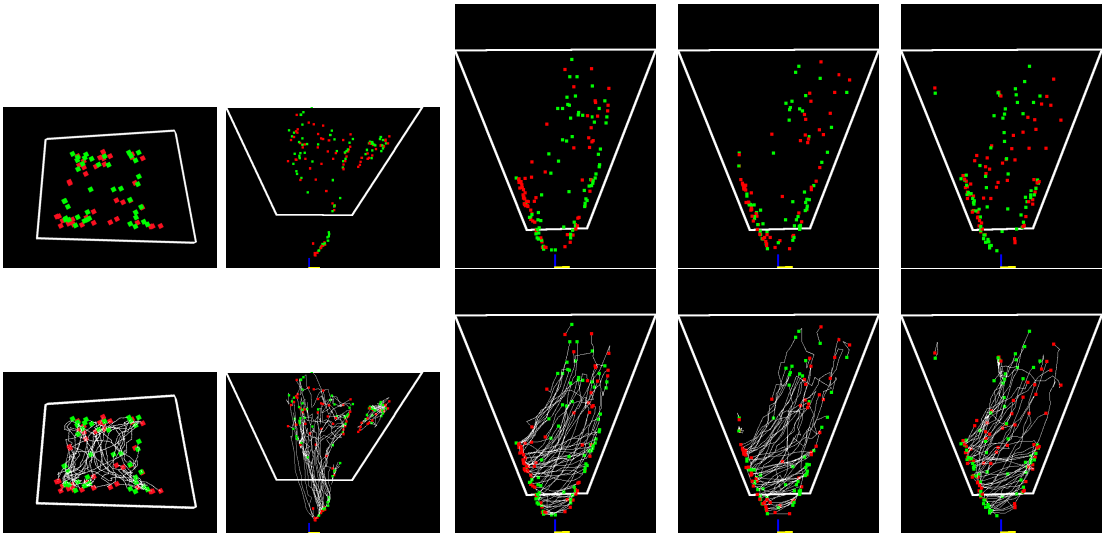
Symbol	Meaning
G	the weighted bipartite graph $G = (V, E)$
E	the weighted bipartite graph G's edges, each edge is a match from a pedestrian x to a track y
$\hat{E}$	a subset of the weighted bipartite graph G's edges
e_{xy}	$e_{xy} \in E$ and *possibly* an element of $\hat{E}$, it is a match from a pedestrian x to a track y
e_{xy}^{w}	the weighting associated with e_{xy}
p_x	pedestrian number x in frame i
$p_x^{3d^i}$	the position of the centre of mass of a detected pedestrian's 3D head region orthographically projected onto the groundplane in frame i
$p_x^{max^i}$	the maximum height above the groundplane of the pedestrian in frame i
$p_x^{min^i}$	the minimum height above the groundplane of the pedestrian in frame i
t_y	track number y in frame $i-1$
$t_y^{c^{i-1}}$	the set of HSV colour values of all foreground points belonging to the pedestrian in frame $i-1$
$t_y^{3d^{i-1}}$	the position of the centre of mass of a tracked pedestrian's 3D head region orthographically projected onto the groundplane in frame $i-1$
$t_y^{max^{i-1}}$	the maximum height above the groundplane of the pedestrian in frame $i-1$
$t_y^{min^{i-1}}$	the minimum height above the groundplane of the pedestrian in frame $i-1$
$t_y^{c^{i-1}}$	the set of HSV colour values of all foreground points belonging to the pedestrian in frame $i-1$
$t_y^{n^{i-1}}$	the number of frames that the track has existed for
$t_y^{v^{i-1}}$	the velocity of the track in frame $i-1$
$t_y^{3d^i}$	the extrapolated position of the track in frame i
$t_y^{s^{i-1}}$	the track state, which is either *walking*, St^w, *accelerating*, St^a, or *standing*, St^s

Table B.2: Pedestrian tracking symbols overview.

APPENDIX C

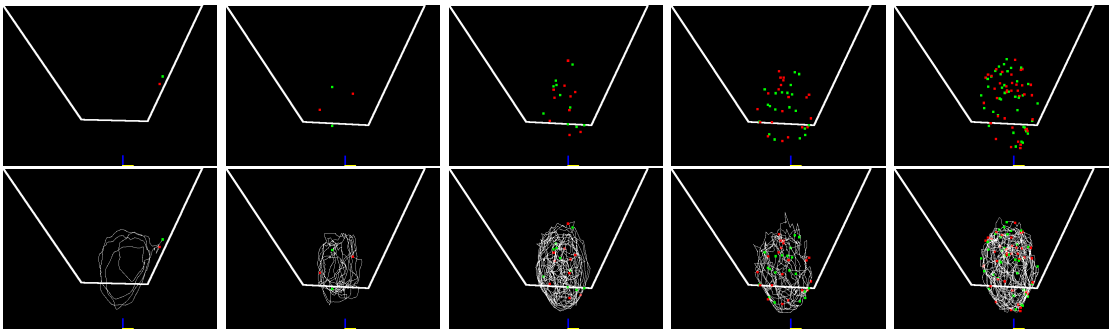
Track Paths

The full paths traversed in the dataset sequences can be seen in the figures C.1 and C.2. Each sequence in these figures has two images presented in plan-view format. The image in the top row presents the positions in the scene where tracks were begun (in green) and terminated (in red). The bottom row image then links up each start and end position by a white line that details the exact path taken by a track. An interesting observation within these figures is the positions in the scenes where paths are lost. For example in figure C.1(b) there is a region (in the top right) where tracks are always lost. This region occurs due to pedestrians passing behind a pillar in the *Corridor* sequence for two or more frames, therefore their tracks are always lost. A second interesting occurrence exists in the top left of figure C.1(e) where a single track exists. This track occurs in the *Grafton* 3 sequence, and as such depicts pedestrian movement from *inside* a building. However, this occurrence is impossible as the camera does not have a view of this area. Inspection of this track reveals it occurs due to a reflection of a pedestrian on a window in the scene. The disparity of this pedestrian therefore places them behind the window, whereas in reality the pedestrian's position is out of the camera's field of view, towards the mid-right of the scene. Finally in the *Vicon* sequences of figures C.2(c), (d) and (e) it should be noticed that on occasion the path of a pedestrian is momentarily outside of the 5.5×3.15 metre elliptical area. This occurs due to the extrapolation of a pedestrian whilst they are occluded, then when the pedestrian becomes un-occluded the track returns to within the Vicon tracking area.



(a) (b) (c) (d) (e)

Figure C.1: *2D* sequence tracks; (a) *Overhead* sequence; (b) *Corridor* sequence; (c) *Grafton* 1 sequence; (d) *Grafton* 2 sequence; (e) *Grafton* 3 sequence; (Row 1) Start (green) and end (red) track positions; (Row 2) Full track paths.



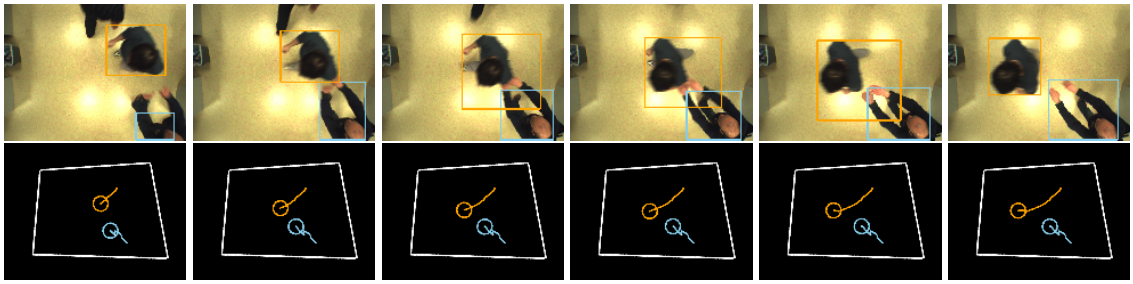
(a) (b) (c) (d) (e)

Figure C.2: *3D* sequence tracks; (a) *Vicon* 1 sequence; (b) *Vicon* 2 sequence; (c) *Vicon* 4 sequence; (d) *Vicon* 8_A sequence; (e) *Vicon* 8_B sequence; (Row 1) Start (green) and end (red) track positions; (Row 2) Full track paths.

APPENDIX D

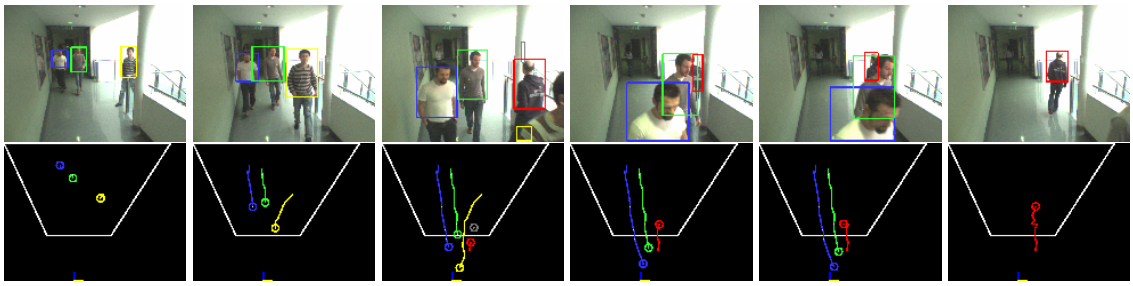
Additional Tracking Results

Further illustrative examples of the final pedestrian detection and tracking system from challenging scenarios are presented in figures D.1-D.5.



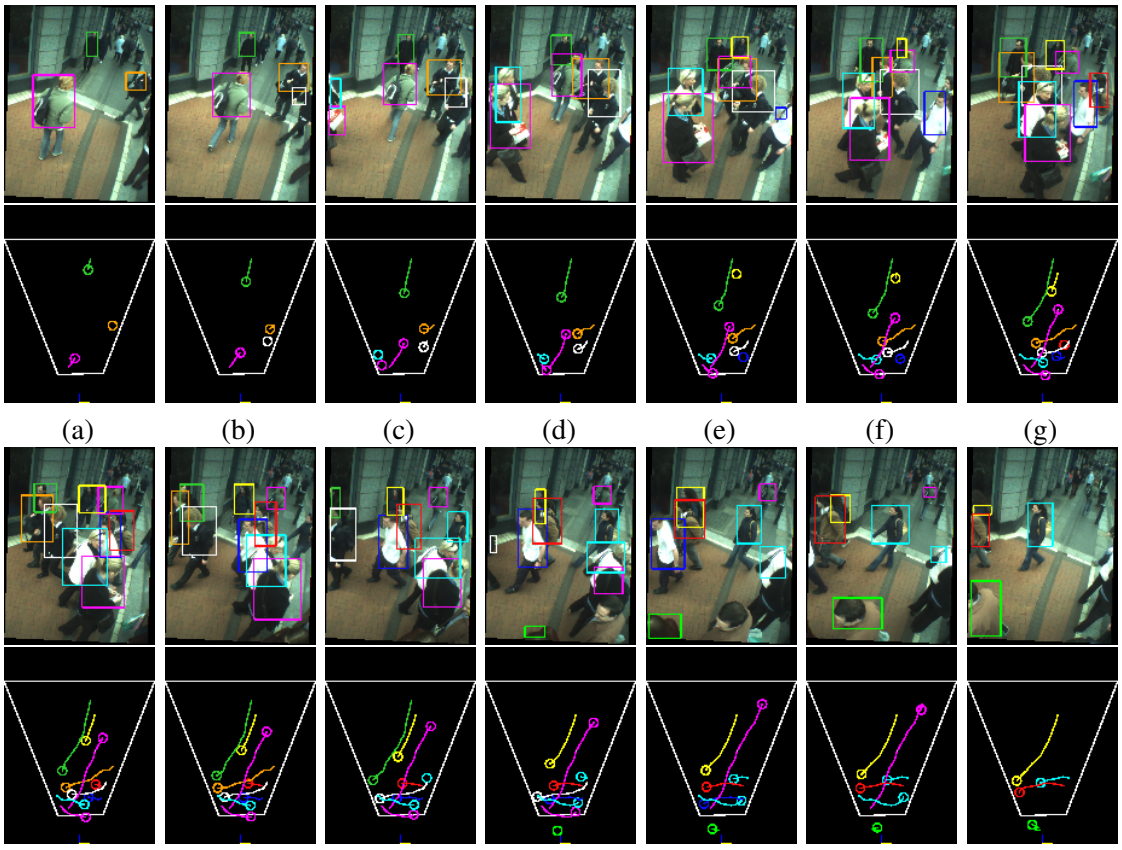
(a) (b) (c) (d) (e) (f)

Figure D.1: *Overhead* sequence. Frame numbers between (a) 185 - (f) 190.



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

Figure D.2: *Corridor* sequence. Frame numbers; (a) 217; (b) 225; (c) 234; (d) 238; (e) 239; (f) 243;



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

(h) (i) (j) (k) (l) (m) (n)

Figure D.3: *Grafton* sequence 1. Frame numbers between (a) 6 - (n) 19.

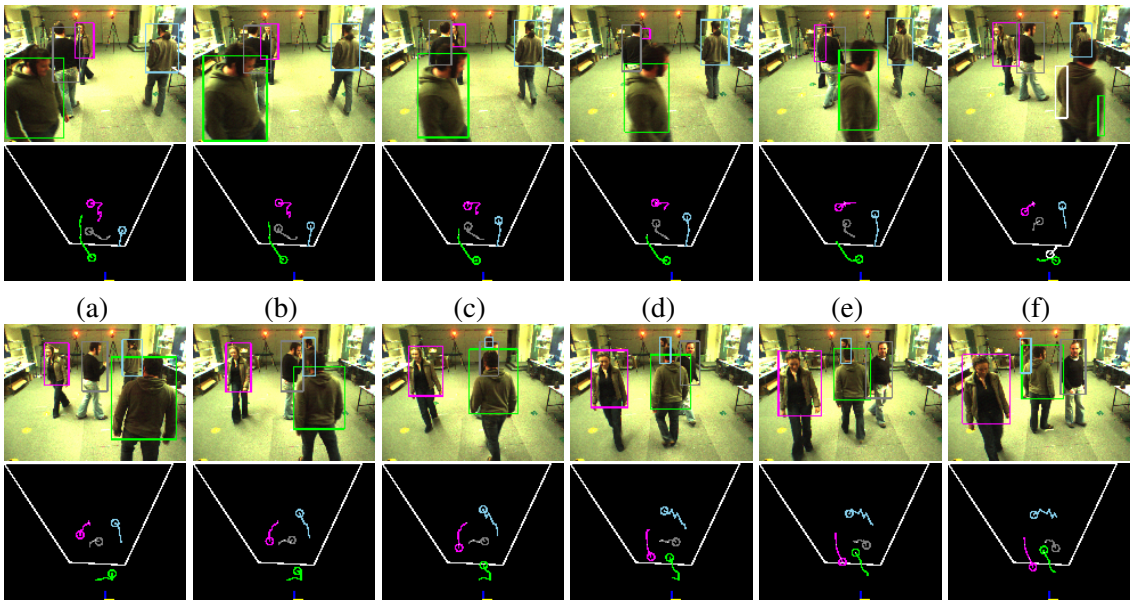


Figure D.4: *Vicon* 4 sequence. Even frame numbers between (a) 54 - (l) 76.

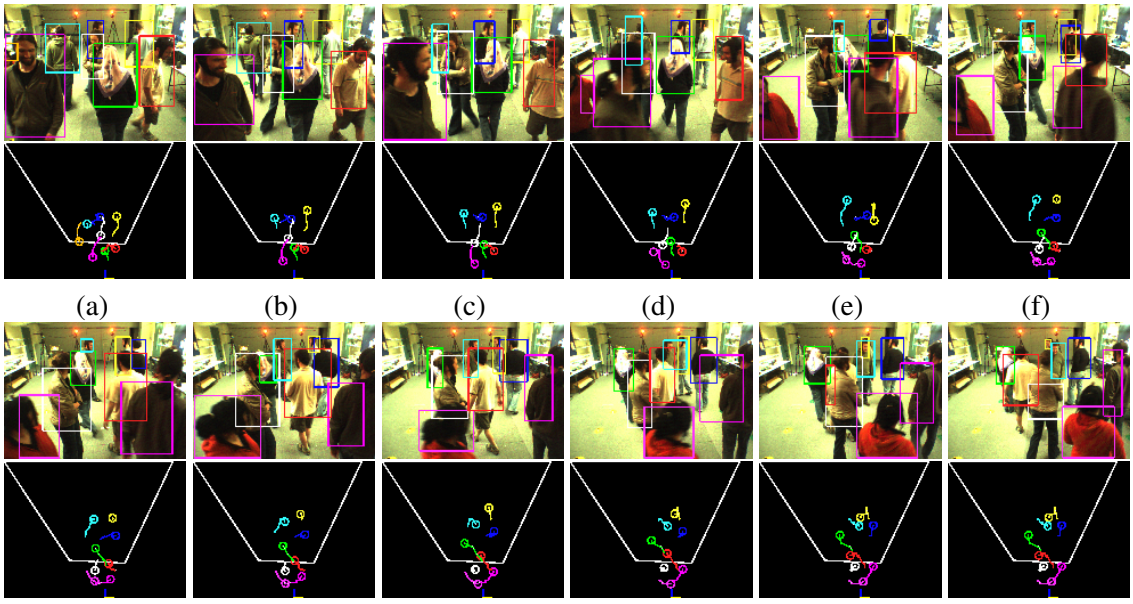


Figure D.5: *Vicon* 8_A sequence. Even frame numbers between (a) 200 - (l) 220.

Bibliography

[1] M. Bertozzi, E. Binelli, A. Broggi, and M. Del Rose. Stereo vision-based approaches for pedestrian detection. In *IEEE International Workshop on Object Tracking and Classification in and Beyond the Visible Spectrum*, 2005.

[2] J. Garcia, N. Da Vitoria Lobo, M. Shah, and J. Feinstein. Automatic detection of heads in colored images. In *Canadian Conference on Computer and Robot Vision*, pages 276–281, 2005.

[3] A. Broggi, M. Bertozzi, A. Fascioli, and M. Sechi. Shape-based pedestrian detection. In *IEEE Intelligent Vehicles Symposium*, pages 215–220, 2000.

[4] T. Zhao and R. Nevatia. Bayesian human segmentation in crowded situations. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 459–466, 2003.

[5] D. Beymer and K. Konolige. Real-time tracking of multiple people using continuous detection. In *International Conference on Computer Vision*, 1999.

[6] D.M. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *IEEE International Conference on Computer Vision*, volume 1, pages 87–93, 1999.

[7] D.M. Gavrila and L.S. Davis. 3d model-based tracking of human upper body movement: a multi-view approach. In *IEEE International Symposium on Computer Vision*, pages 253–258, 1995.

[8] Q. Delamarre and O. Faugeras. 3d articulated models and multi-view tracking with silhouettes. In *International Conference on Computer Vision*, volume 2, pages 716–721, 1999.

[9] A. Baumberg. *Learning Deformable Models for Tracking Human Motion*. PhD thesis, University of Leeds, 1995.

[10] A. Shashua, Y. Gdalyahu, and G. Hayun. Pedestrian detection for driving assistance systems: single-frame classification and system level performance. In *IEEE Intelligent Vehicles Symposium*, pages 1–6, 2004.

[11] P.I. Alonso, F.D. Llorca, M.A. Sotelo, L.M. Bergasa, R.P. de Toro, J. Nuevo, M. OcañaOcana, and M.A. García Garrido. Combination of feature extraction methods for svm pedestrian detection. In *IEEE Transactions on Intelligent Transportation Systems*, 2007.

[12] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis and Machine Vision, Second Edition*. PWS Publishing, 1999.

[13] F. Suard, V. Guigue, A. Rakotomamonjy, and A. Benshrair. Pedestrian detection using stereo-vision and graph kernels. In *IEEE Intelligent Vehicles Symposium*, pages 267–272, 2005.

[14] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easyliving. In *IEEE Workshop on Visual Surveillance*, pages 3–10, 2000.

[15] M. Harville. Stereo person tracking with adaptive plan-view statistical templates. In *Workshop on Statistical Methods in Video Processing*, pages 67–72, 2002.

[16] M. Harville. Stereo person tracking with short and long term plan-view appearance models of shape and color. In *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 522–527, 2005.

[17] K-J. Yoon and I-S. Kweon. Locally adaptive support-weight approach for visual correspondence search. In *British Machine Vision Conference*, pages 924–931, 2005.

[18] A.F. Bobick and S.S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33:181–200, 1999.

[19] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision*, 35:269–293, 1999.

[20] Q. Cai, A. Mitiche, and J.K. Aggarwal. Tracking human motion in an indoor environment. In *International Conference on Image Processing*, volume 1, pages 215–218, 1995.

[21] M. Harville. Stereo person tracking with adaptive plan-view templates of height and occupancy statistics. *International Journal of Computer Vision*, 22:127–142, 2004.

[22] P. Remagnino and G.L. Foresti. Ambient intelligence: A new multidisciplinary paradigm. In *IEEE Transactions on Systems, Man and Cybernetics*, volume 35, pages 1–6, 2005.

[23] M. Vallée, F. Ramparany, and L. Vercouter. A multi-agent system for dynamic service composition in ambient intelligence environments. In *International Conference on Pervasive Computing*, pages 157–182, 2005.

[24] C. Wöhler, J.K. Aulanf, T. Portner, and U. Franke. A time delay neural network algorithm for real-time pedestrian recognition. In *International Conference on Intelligent Vehicle*, pages 247–252, 1998.

[25] C. Curio, J. Edelbrunner, T. Kalinke, C. Tzomakas, and W. Von Seelen. Walking pedestrian recognition. In *IEEE Transactions on Intelligent Transportation Systems*, volume 1, pages 155–163, 2000.

[26] D.M. Gavrila, J. Giebel, and S. Munder. Vision-based pedestrian detection: the protector+ system. In *IEEE Intelligent Vehicles Symposium*, pages 13–18, 2004.

[27] W. Hu, , T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviours. In *IEEE Transactions on Systems, Man, and Cybernetics - Part C*, volume 34, pages 334–352, 2004.

[28] A. Rourke and M.G.H. Bell. An image-processing system for pedestrian data collection. In *International Conference on Road Traffic Monitoring and Control*, pages 123–126, 1994.

[29] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 878–885, 2005.

[30] J. Leskovec and S. Sentjo. Detection of human bodies using computer analysis of a sequence of stereo images. In *European Union Contest for Young Scientists*, 1999.

[31] B. Maurin, O. Masoud, and N.P. Papanikolopoulos. Camera surveillance of crowded traffic scenes. In *ITS America Annual Meeting*, pages 28–55, 2002.

[32] A. Broggi, A. Fascioli, P. Grisleri, T. Graf, and M. Meinecke. Model-based validation approaches and matching techniques for automotive vision based pedestrian detection. In *IEEE International Workshop on Object Tracking and Classification in and Beyond the Visible Spectrum*, 2005.

[33] H. Nanda and L.S. Davis. Probabilistic template based pedestrian detection in infrared videos. In *IEEE Intelligent Vehicle Symposium*, volume 1, pages 15–20, 2002.

[34] F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi. Pedestrian detection using infrared images and histograms of oriented gradients. In *IEEE Intelligent Vehicles Symposium*, pages 206–212, 2006.

[35] Q. Cai and J.K. Aggarwal. Tracking human motion using multiple cameras. In *International Conference on Pattern Recognition*, volume 3, pages 68–72, 1996.

[36] K. Sato and J.K. Aggarwal. Temporal spatio-velocity transform and its application to tracking and interaction. *Computer Vision and Image Understanding*, 96:100–128, 2004.

[37] C. BenAbdelkader, R. Cutler, and L.S. Davis. Stride and cadence as a biometric in automatic person identification and verification. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 357–362, 2002.

[38] Y. Ran, Q. Zheng, I. Weiss, L.S. Davis, W. Abd-Almageed, and L. Zhao. Pedestrian classification from moving platforms using cyclic motion pattern. In *IEEE International Conference on Image Processing*, volume 2, pages 854–857, 2005.

[39] S.A. Niyogi and E.H. Adelson. Analyzing and recognizing walking figures in xyt. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 469–474, 1994.

[40] O. Javed and M. Shah. Tracking and object classification for automated surveillance. In *European Conference on Computer Vision*, pages 343–357, 2002.

[41] J. Black and T.J. Ellis. Multi camera image tracking. In *Image and Vision Computing*, volume 24, pages 1256–1267, 2006.

[42] L. Snidaro, C. Micheloni, and C. Chiavedale. Video security for ambient intelligence. In *IEEE Transactions on Systems, Man and Cybernetics, Part A*, volume 35, pages 133–144, 2005.

[43] K. Uchida, J. Miura, and Y. Shirai. Tracking multiple pedestrians in crowd. In *Workshop of Machine Vision and its Applications*, pages 533–536, 2000.

[44] O. Masoud and N.P. Papanikolopoulos. A novel method for tracking and counting pedestrians in real-time using a single camera. In *IEEE Transactions on Vehicular Technology*, volume 50, pages 1267–1278, 2001.

[45] L.Q. Xu and D.C. Hogg. Neural networks in human motion tracking - an experimental study. *Image and Vision Computing*, 15:607–615, 1997.

[46] I. Haritaoglu, D. Harwood, and L.S. Davis. W^4: Who? when? where? what? a real time system for detecting and tracking people. In *International Conference on Face and Gesture Recognition*, pages 222–227, 1998.

[47] A.J. Lipton, H. Fujiyoshi, and R.S. Patil. Moving target classification and tracking from real-time video. In *IEEE Workshop on Applications of Computer Vision*, pages 8–14, 1998.

[48] A.R. Madabhushi and J.K. Aggarwal. Using head movement to recognize human activity. In *International Conference on Pattern Recognition*, volume 4, pages 698–701, 2000.

[49] M-T. Yang, Y-C. Shih, and S-C. Wang. People tracking by integrating multiple features. In *International Conference on Pattern Recognition*, volume 4, pages 929–932, 2004.

[50] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti. Improving shadow suppression in moving object detection with hsv color information. In *IEEE Intelligent Transportation System Conference*, pages 334–339, 2001.

[51] T.B. Moeslund and E. Granum. Multiple cues used in model-based human motion capture. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 362–367, 2000.

[52] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting objects, shadows and ghosts in video streams by exploiting color and motion information. In *International Conference on Image Analysis and Processing*, pages 360–365, 2001.

[53] N.T. Siebel and S.J. Maybank. Fusion of multiple tracking algorithms for robust people tracking. In *European Conference on Computer Vision*, pages 373–387, 2002.

[54] H-T. Chen, H-H. Lin, and T-L. Liu. Multi-object tracking using dynamical graph matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, volume 2, pages 210–217, 2001.

[55] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19, pages 780–785, 1997.

[56] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, and S. Maybank. Principal axis-based correspondence between multiple cameras for people tracking. In *Pattern Analysis and Machine Intelligence*, volume 28, pages 663–671, 2006.

[57] S. Khan and M. Shah. Tracking people in presence of occlusion. In *Asian Conference on Computer Vision*, 2000.

[58] T. Zhao, R. Nevatia, and F. Lv. Segmentation and tracking of multiple humans in complex situations. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 194–201, 2001.

[59] H. Wang and D. Suter. A re-evaluation of mixture of gaussian background modeling. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1017–1020, 2005.

[60] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 246–252, 1999.

[61] M. Harville, G. Gordon, and J. Woodfill. Adaptive background subtraction using color and depth. In *IEEE International Conference on Image Processing*, volume 3, pages 90–93, 2001.

[62] A. Datta, M. Shah, and N. Da Vitoria Lobo. Person-on-person violence detection in video data. In *IEEE International Conference on Pattern Recognition*, volume 1, pages 433–438, 2002.

[63] S.M. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *European Conference on Computer Vision*, pages 133–146, 2006.

[64] A. Elgammal, D. Harwood, and L.S. Davis. Non-parametric model for background subtraction. In *European Conference on Computer Vision*, pages 751–767, 2000.

[65] H. Nanda, C. BenAbdelkader, and L.S. Davis. Modelling pedestrian shapes for outlier detection: A neural net based approach. In *IEEE Intelligent Vehicle Symposium*, pages 428–433, 2003.

[66] S. Krotosky and M. Trivedi. Multimodal stereo image registration for pedestrian detection. In *IEEE Intelligent Transportation Systems Conference*, pages 109–114, 2006.

[67] K. Kim, D. Harwood, and L.S. Davis. Background updating for visual surveillance. In *International Symposium on Visual Computing*, pages 337–346, 2005.

[68] K. Kim and L.S. Davis. Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In *European Conference on Computer Vision (Part 3)*, pages 98–109, 2006.

[69] S.S. Intille, J.W. Davis, and A.F. Bobick. Real-time closed-world tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 697–703, 1997.

[70] S.J. McKenna, S. Jabri, Z. Duric, and H. Wechsler. Tracking interacting people. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 348–454, 2000.

[71] X. Zhang and G. Sexton. A new method for pedestrian counting. In *International Conference on Image Processing and its Applications*, pages 208–212, 1995.

[72] P. Beardsley and E. Bourrat. Wheelchair detection using stereo vision. Technical report, MERL, 2002.

[73] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. *International Journal of Computer Vision*, 37:175–185, 2000.

[74] D.R.P. Gibson, B. Ling, M. Zeifman, S. Dong, and U. Venkataraman. Multipedestrian tracking. In *Public Roads*, volume 69, 2006.

[75] R. Luo and Y. Guo. Real-time stereo tracking of multiple moving heads. In *IEEE ICCV Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 55–60, 2001.

[76] L. Zhao and C. Thorpe. Recursive context reasoning for human detection and parts identification. In *IEEE Workshop on Human Modeling, Analysis, and Synthesis*, 2000.

[77] T. Darrell, D. Demirdjian, N. Checka, and P. Felzenszwalb. Plan-view trajectory estimation with dense stereo background models. In *IEEE International Conference on Computer Vision*, volume 2, pages 628–635, 2001.

[78] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *IEEE International Conference on Computer Vision*, pages 255–261, 1999.

[79] A.W. Senior. Tracking with probabilistic appearance models. In *ECCV workshop on Performance Evaluation of Tracking and Surveillance Systems*, pages 48–55, 2002.

[80] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills. Recovering motion fields: An evaluation of eight optical flow algorithms. In *British Machine Vision Conference*, pages 195–204, 1998.

[81] B.K.P. Horn and B.G. Rhunck. Determining optical flow: a retrospective. *Artificial Intelligence*, 59:81–87, 1993.

[82] H. Mori, N.M. Charkari, and T. Matsushita. On-line vehicle and pedestrian detections based on sign pattern. In *IEEE Transactions on Industrial Electronics*, volume 41, pages 384–391, 1994.

[83] L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. In *International Conference on Pattern Recognition*, volume 36, pages 585–601, 2003.

[84] H. Tsutsui, J. Miura, and Y. Shirai. Optical flow-based person tracking by multiple cameras. In *IEEE Conference Multisensor Fusion and Integration in Intelligent Systems*, pages 91–96, 2001.

[85] A.T. Ali and E.L. Dagless. Vehicle and pedestrian detection and tracking. In *IEE Colloquium on Image Analysis for Transport Applications*, pages 48–54, 1990.

[86] G. Appenzeller, Y. Kunii, and H. Hashimoto. A low-cost real-time stereo vision system for looking at people. In *IEEE International Symposium on Industrial Electronics*, volume 3, pages 767–772, 1997.

[87] A.E Elgammal and L.S. Davis. Probabilistic framework for segmenting people under occlusion. In *IEEE International Conference on Computer Vision*, volume 2, pages 145–152, 2001.

[88] O. Masoud and N.P. Papanikolopoulos. A robust real-time multi-level model-based pedestrian tracking system. In *ITS America Annual Meeting*, pages 47–69, 1997.

[89] O. Masoud and N.P. Papanikolopoulos. Robust pedestrian tracking using a model-based approach. In *IEEE Conference on Intelligent Transportation Systems*, pages 338–343, 1997.

[90] K. Terada, D. Yoshida, S. Oe, and J. Yamaguchi. A counting method of the number of passing people using a stereo camera. In *IEEE Conference on Industrial Electronics*, volume 3, pages 1318–1323, 1999.

[91] I. Haritaoglu, D. Beymer, and M. Flickner. Ghost3d: Detecting body posture and parts using stereo. In *Workshop on Motion and Video Computing*, pages 175–180, 2002.

[92] M.R. Crabtree. Smart pedestrian counter system (specs). In *International Conference on Road Transport Information and Control*, pages 100–104, 2002.

[93] A. Taleb-Ahmed, N. Ducrocq, and G. Tilmanp. Positioning sensors, video tool for counting pedestrians. In *IEEE Conference on Systems, Man and Cybernetics*, volume 4, pages 612–617, 1999.

[94] R.T. Collins, A.J. Lipton, H. Fujiyoshi, and T. Kanade. Algorithms for cooperative multisensor surveillance. In *Proceedings of the IEEE*, pages 1456–1477, 2001.

[95] X. Yuan, Y-J. Lu, and S. Sarrif. A computer vision system for measurement of pedestrian volume. In *IEEE Region 10 Conference on Computer, Communication, Control and Power Engineering*, volume 2, pages 1046–1049, 1993.

[96] K. Sato and J.K. Aggarwal. Tracking and recognizing two-person interactions in outdoor image sequences. In *IEEE Workshop on Multi-Object Tracking*, pages 87–94, 2001.

[97] T. Sogo, H. Ishiguro, and M.M. Trivedi. N-ocular stereo for real-time human tracking. In *Panoramic vision: sensors, theory, and applications book contents*, pages 359–375, 2001.

[98] D.M. Gavrila and J. Giebel. Shape-based pedestrian detection and tracking. In *IEEE Intelligent Vehicles Symposium*, volume 1, pages 8–14, 2002.

[99] P. Remagnino, A. Baumberg, T. Grove, D. Hogg, T. Tan, A. Worrall, and K. Baker. An integrated traffic and pedestrian model based vision system. In *British Machine Vision Conference*, volume 2, pages 380–389, 1997.

[100] F. De la Torre, C. Vallespi, P.E. Rybski, M. Veloso, and T. Kanade. Learning to track multiple people in omnidirectional video. In *IEEE International Conference on Robotics and Automation*, pages 4150–4155, 2005.

[101] L. Zhao and L.S. Davis. Closely coupled object detection and segmentation. In *IEEE International Conference on Image Processing*, pages 454–461, 2005.

[102] L. Zhao and C. Thorpe. Stereo and neural network-based pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems*, 1:148–154, 2000.

[103] D.M. Gavrila and L.S. Davis. 3d model-based tracking of humans in action: a multi-view approach. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 73–80, 1996.

[104] Q. Delamarre and O. Faugeras. 3d articulated models and multi-view tracking with physical forces. In *Computer Vision and Image Understanding*, volume 81, pages 328–357, 2001.

[105] K. Hayashi, M. Hashimoto, K. Sumi, and K. Sasakawa. Multiple-person tracker with a fixed slanting stereo camera. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 681–686, 2004.

[106] X. Liu, P.H. Tu, J. Rittscher, A. Perera, and N. Krahnstoever. Detecting and counting people in surveillance applications. In *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 306–311, 2005.

[107] V. Philomin, R. Duraiswami, and L.S. Davis. Pedestrian tracking from a moving vehicle. In *IEEE Intelligent Vehicles Symposium*, pages 350–355, 2000.

[108] U. Franke, D. Gavrila, S. Görzig, F. Lindner, F. Paetzold, and C. Wöhler. Autonomous driving goes downtown. *IEEE Intelligent Systems*, 13:40–48, 1998.

[109] O. Sidla, Y. Lypetskyy, N. Brandle, and S. Seer. Pedestrian detection and tracking for counting applications in crowded situations. In *IEEE International Conference on Video and Signal Based Surveillance*, pages 70–75, 2006.

[110] C. Kiefer. Qualitative and quantitative evaluation of the reading people tracker. Master's thesis, ETH Zurich, 2004.

[111] C. Papageorgiou, T. Evgeniou, and T. Poggio. A trainable pedestrian detection system. In *IEEE Intelligent Vehicles Symposium*, pages 241–246, 1998.

[112] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 193–199, 1997.

[113] M.A.l. Sotelo, I. Parra, D. Fernandez, and E. Naranjo. Pedestrian detection using svm and multi-feature combination. In *IEEE Intelligent Transportation Systems Conference*, pages 103–108, 2006.

[114] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 886–893, 2005.

[115] P. Viola, M.J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *IEEE International Conference on Computer Vision*, volume 2, pages 734–741, 2003.

[116] S.L. Phung and A. Bouzerdoum. Detecting people in images - an edge density approach. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 1229–1232, 2007.

[117] A. Micilotta, E. Ong, and R. Bowden. Detection and tracking of humans by probabilistic body part assembly. In *British Machine Vision Conference*, volume 1, pages 419–428, 2005.

[118] R.E. Schapire, Y. Freund, P. Bartlett, and W.S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.

[119] P. Viola and M.J. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE International Conference on Computer Vision*, volume 1, pages 511–518, 2001.

[120] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *European Conference on Computer Vision*, volume 1, pages 69–81, 2004.

[121] S. Ayers and M. Shah. Monitoring human behaviour in an office environment. In *IEEE Computer Society Workshop: The Interpretation of Visual Motion*, pages 65–72, 1998.

[122] R. Munoz-Salinas, E. Aguirre, M. Garcia-Silvente, and A. Gonzalez. People detection and tracking through stereo vision for human-robot interaction. In *Lectures Notes on Artificial Intelligence*, pages 337–346, 2005.

[123] R. Cutler and L.S. Davis. Robust real-time periodic motion detection: Analysis and applications. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 781–796, 1999.

[124] I. Haritaoglu, R. Cutler, D. Harwood, and L.S. Davis. Backpack: Detection of people carrying objects using silhouettes. *Computer Vision and Image Understanding*, 81:385–397, 2001.

[125] C-J. Pai, H-R. Tyan, Y-M. Liang, H-Y.M. Liao, and S-W. Chen. Pedestrian detection and tracking at crossroads. In *International Conference on Image Processing*, volume 2, pages 101–104, 2003.

[126] B. Heisele and C. Wöhler. Motion-based recognition of pedestrians. In *International Conference on Pattern Recognition*, volume 2, pages 1325–1330, 1998.

[127] S. Yasutomi and H. Mori. A method for discriminating of pedestrian based on rhythm. In *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, volume 2, pages 988–995, 1994.

[128] A. Azarbayejani and A. Pentland. Real-time self-calibrating stereo person tracking using 3-d shape estimation from blob features. In *International Conference on Pattern Recognition*, volume 3, pages 627–632, 1996.

[129] H. Elzein, S. Lakshmanan, and P. Watta. A motion and shape-based pedestrian detection algorithm. In *IEEE Intelligent Vehicles Symposium*, pages 500–504, 2003.

[130] B. Maurin, O. Masoud, and N.P. Papanikolopoulos. Computer vision algorithms for monitoring crowded scenes. *IEEE Robotics and Automation Magazine*, pages 29–36, 2003.

[131] J. Batista. Tracking pedestrians under occlusion using multiple cameras. In *Image Analysis and Recognition*, pages 552–562, 2004.

[132] M. Bertozzi, A. Broggi, A. Fascioli, A. Tibaldi, R. Chapuis, and F. Chausse. Pedestrian localization and tracking system with kalman filtering. In *IEEE Intelligent Vehicles Symposium*, pages 584–589, 2004.

[133] I. Haritaoglu, D. Harwood, and L.S. Davis. Hydra: multiple people detection and tracking using silhouettes. In *International Conference on Image Analysis and Processing*, pages 280–285, 1999.

[134] Y. Raja, S.J. McKenna, and S. Gong. Tracking and segmenting people in varying lighting conditions using colour. In *International Conference on Face and Gesture Recognition*, pages 228–233, 1998.

[135] N. Thome and S. Miguet. A robust appearance model for tracking human motions. In *IEEE International Conference on Advanced Video and Signal-Based Surveillance*, pages 528–533, 2005.

[136] A. Azarbayejani, C. Wren, and A. Pentland. Real-time 3-d tracking of the human body. In *IMAGE'COM*, 1996.

[137] A. Mittal and L.S. Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo. In *European Conference on Computer Vision*, volume 1, pages 18–36, 2002.

[138] H-D. Yang and S-W. Lee. Multiple pedestrian detection and tracking based on weighted temporal texture features. In *International Conference on Pattern Recognition*, volume 4, pages 248–251, 2004.

[139] L. Iocchi and R.C. Bolles. Integrating plan-view tracking and color-based person models for multiple people tracking. In *IEEE International Conference on Image Processing*, volume 3, pages 872–875, 2005.

[140] S. Bahadori, G. Grisetti, L. Iocchi, G.R. Leone, and D. Nardi. Real-time tracking of multiple people through stereo vision. In *IEE International Workshop on Intelligent Environments*, pages 252–259, 2005.

[141] I. Haritaoglu, D. Harwood, and L.S. Davis. An appearance-based body model for multiple people tracking. In *International Conference on Pattern Recognition*, volume 4, pages 184–187, 2000.

[142] I. Haritaoglu, D. Harwood, and L.S. Davis. W^4s: A real time system for detecting and tracking people in 2.5 d. In *European Conference on Computer Vision*, pages 877–892, 1998.

[143] M. Bertozzi, A. Broggi, R. Chapuis, F. Chausse, A. Fascioli, and A. Tibaldi. Shape-based pedestrian detection and localization. In *IEEE International Conference on Intelligent Transportation Systems*, pages 328–333, 2003.

[144] K.C. Fuerstenberg and V. Willhoeft. Object tracking and classification using laserscanners - pedestrian recognition in urban environment. In *IEEE International Conference on Intelligent Transport Systems*, pages 451–453, 2001.

[145] K.C. Fuerstenberg, K.C.J. Dietmayer, and V. Willhoeft. Pedestrian recognition in urban traffic using a vehicle based multilayer laserscanner. In *IEEE Intelligent Vehicles Symposium*, volume 1, pages 31–35, 2002.

[146] S. Bahadori, L. Iocchi, D. Nardi, and G.P. Settembre. Stereo vision based human body detection from a localized mobile robot. In *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 499–504, 2005.

[147] D.M. Gavrila, U. Franke, S. Görzig, and C. Wöhler. Real-time vision for intelligent vehicles. *IEEE Instrumentation and Measurement Magazine*, 4:22–27, 2001.

[148] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision Second Edition*. Cambridge University Press, 2003.

[149] D.A. Forsythe and J. Ponce. *Computer Vision - A Modern Approach*. Prentice Hall, 2003.

[150] S. Birchfield. An introduction to projective geometry (for computer vision).

[151] R. Mohr and B. Triggs. Projective geometry for image analysis. In *International Symposium of Photogrammetry and Remote Sensing*, pages 1532–1534, 1996.

[152] M. Bergtholdt. Auto-calibration with convex constraints. Master's thesis, University of Mannheim, 2002.

[153] R.I. Hartley. Theory and practice of projective rectification. *International Journal of Computer Vision,*, 35:115 – 127, 1999.

[154] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12:16–22, 2000.

[155] D. Oram. Rectification for any epipolar geometry. In *British Machine Vision Conference*, 1988.

[156] U.R. Dhond and J.K. Aggarwal. A linear method for trinocular rectification. In *IEEE International Conference on Robotics and Automation*, pages 2045–2050, 1990.

[157] D. Weinshall, M. Werman, and A. Shashua. Shape tensors for efficient and learnable indexing. In *IEEE Workshop on Representation of Visual Scenes*, pages 58–65, 1995.

[158] K.M. Cheung, T. Kanade, J. Bouguet, and M. Holler. A real time system for robust 3d voxel reconstruction of human motions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 714–720, 2000.

[159] A. Lopez, C. Canton-Ferrer, and J.R. Casas. Multi-person 3d tracking with particle filters on voxels. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 913–916, 2007.

[160] R. Labayrade, D. Aubert, and J-P. Tarel. Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation. In *IEEE Intelligent Vehicle Symposium*, volume 2, pages 646–651, 2002.

[161] G. Grubb, A. Zelinsky, L. Nilsson, and M. Rilbe. 3d vision sensing for improved pedestrian safety. In *IEEE Intelligent Vehicles Symposium*, pages 19–24, 2004.

[162] M. Bertozzi, A. Broggi, M. Felisa, G. Vezzoni, and M. Del Rose. Low-level pedestrian detection by means of visible and far infra-red tetra-vision. In *IEEE Intelligent Vehicles Symposium*, pages 206–212, 2006.

[163] H. Wang, Q. Chen, and W. Cai. Shape-based pedestrian/bicyclist detection via onboard stereo vision. In *IMACS Multiconference on Computational Engineering in Systems Applications*, pages 1776–1780, 2006.

[164] M.A. Keck, J.W. Davis, and A. Tyagi. Tracking mean shift clustered point clouds for 3d surveillance. In *ACM International Workshop on Video Surveillance and Sensor Networks*, pages 187–194, 2006.

[165] P. Kelly, P. Beardsley, E. Cooke, N.E. O'Connor, and A.F. Smeaton. Detecting shadows and low-lying objects in indoor and outdoor scenes using homographies. In *IEE International Conference on Visual Information Engineering*, pages 393–400, 2005.

[166] D. Beymer. Person counting using stereo. In *Workshop on Human Motion*, pages 127–131, 2000.

[167] D. Beymer and K. Konolige. Tracking people from a mobile platform. In *Workshop on Reasoning with Uncertainty in Robotics*, 2001.

[168] S. Bahadori, L. Iocchi, R. Leone, D. Nardi, and L. Scozzafava. Real-time people localization and tracking through fixed stereo vision. In *Applied Intelligence*, volume 26, pages 83–97, 2007.

[169] http://www.ptgrey.com/products/digiclops/index.html.

[170] http://www.ptgrey.com.

[171] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision,*, 47:7–42, 2002.

[172] C. Zitnick and T. Kanade. A cooperative algorithm for stereo matching and occlusion detection. Technical report, Robotics Institute, Carnegie Mellon University, 1999.

[173] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 195–202, 2003.

[174] Y. Wei and L. Quan. Region-based progressive stereo matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 106–113, 2004.

[175] M. Lhuillier and L. Quan. Robust dense matching using local and global geometric constraints. In *International Conference on Pattern Recognition*, volume 1, pages 968–972, 2000.

[176] O. Faugeras, B. Hotz, H. Mathieu, T. Viéville, Z. Zhang, P. Fua, E. Théron, L. Moll, G. Berry, J. Vuillemin, P. Bertin, and C. Proy. Real time correlation-based stereo: algorithm, implementations and applications. Technical report, INRIA, 1993.

[177] J-I. Park and S. Inoue. Hierarchical depth mapping from multiple cameras. In *International Conference on Image Analysis and Processing*, volume 1, pages 685–692, 1997.

[178] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: theory and experiment. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 16, pages 920–932, 1994.

[179] R. Shukla, H. Radha, and M. Vetterli. Disparity dependent segmentation based stereo image coding. In *International Conference on mage Processing*, volume 1, pages 757–760, 2003.

[180] J-C. Kim and J-H. Park. Realistic 3d view generation using deformal stereo matching. In *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, volume 35, pages 768–773, 2004.

[181] K. Mühlmann, D. Maier, J. Hesser, and R. Männer. Calculating dense disparity maps from color stereo images, an efficient implementation. *International Journal of Computer Vision*, 47:30–36, 2002.

[182] C. Kim, K.M. Lee, B.T. Choi, and S.U. Lee. A dense stereo matching using two-pass dynamic programming with generalized ground control points. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1075–1082, 2005.

[183] L. Falkenhagen. Depth estimation from stereoscopic image pairs assuming piecewise continuous surfaces. In *Image Processing for Broadcast and Video Production*, pages 115–127, 1994.

[184] J. Mulligan and K. Daniilidis. Real time trinocular stereo for tele-immersion. In *International Conference on Image Processing*, volume 3, pages 959–962, 2001.

[185] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *European Conference on Computer Vision*, pages 151–158, 1994.

[186] E. Trucco, K. Plakas, N. Brandenburg, P. Kauff, M. Karl, and O. Schreer. Real-time disparity maps for immersive 3-d teleconferencing by hybrid recursive matching and census transform. In *ICCV*, 2001.

[187] R. Collins. A space-sweep approach to true multi-image matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 358–363, 1996.

[188] J. Xiao and M. Shah. Two-frame wide baseline matching. In *IEEE International Conference on Computer Vision*, pages 603–609, 2003.

[189] J. Konrad and Z.D. Lan. Dense disparity estimation from feature correspondences. In *SPIE Stereoscopic Displays and Virtual Reality Systems*, volume 3957, pages 90–101, 2000.

[190] P. Seitz. Using local orientational information as image primitive for robust object recognition. In *Visual Communication and Image Processing IV*, volume SPIE-1199, pages 1630–1639, 1989.

[191] H.K. Nishihara. Practical real-time imaging stereo matcher. *Optical Engineering*, 23:536–545, 1984.

[192] A. Fusiello, V. Roberto, and E. Trucco. Efficient stereo with multiple windowing. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 858–863, 1997.

[193] D. Scharstein and R. Szeliski. Stereo matching with non-linear diffusion. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 343–350, 1996.

[194] R. Yang and M. Pollefeys. Multi-resolution real-time stereo on commodity graphics hardware. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 211–217, 2003.

[195] M Agrawal and L.S. Davis. Window-based, discontinuity preserving stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 66–73, 2004.

[196] F. Busetti. Simulated annealing overview.

[197] R. Szeliski and R. Zabih. An experimental comparison of stereo algorithms. In *International Workshop on Vision Algorithms*, pages 1–19, 1999.

[198] M. Tappen and W. Freeman. Comparison of graph cuts with belief propagation for stereo. In *IEEE International Conference on Computer Vision*, volume 2, pages 900–907, 2003.

[199] O. Veksler. Reducing search space for stereo correspondence with graph cuts. In *British Machine Vision Conference*, volume 2, pages 709–718, 2006.

[200] H. Sunyoto, W. Van der Mark, and D.M. Gavrila. A comparative study of fast dense stereo vision algorithms. In *IEEE Intelligent Vehicles Symposium*, pages 319–324, 2004.

[201] H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 807–814, 2005.

[202] Q-B. Zhang, H-X. Wang, and S. Wei. A new algorithm for 3d projective reconstruction based on infinite homography. In *International Conference on Machine Learning and Cybernetics*, pages 2882–2886, 2003.

[203] K. Kim, T.H. Chalidabhongse, D. Harwood, and L.S. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11:172–185, 2005.

[204] Y. Nakamura, T. Matsuura, K. Satoh, and Y. Ohta. Occlusion detectable stereo-occlusion patterns in camera matrix. In *CVPR*, pages 371–378, 1996.

[205] www.middlebury.edu/stereo.

[206] http://www.vicon.com, 2007.

[207] P. Kelly, E. Cooke, N.E. O'Connor, and A.F. Smeaton. Pedestrian detection using stereo and biometric information. In *International Conference on Image Analysis and Recognition*, pages 802–813, 2006.

[208] http://www.goldennumber.net, 2007.

[209] M. Harville and L. Dalong. Fast, integrated person tracking and activity recognition with plan-view templates from a single stereo camera. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 398–405, 2002.

[210] K. Sobottka and I. Pitas. Extraction of facial regions and features using color and shape information. In *International Conference on Pattern Recognition*, volume 3, pages 421–425, 1996.

[211] P. Kelly, N.E. O'Connor, and A.F. Smeaton. Pedestrian detection in uncontrolled environments using stereo and biometric information. In *ACM International Workshop on Video Surveillance and Sensor Networks*, pages 161–170, 2006.

[212] Z. Galil. Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys*, 18:23–38, 1986.

[213] T. Kailath. The divergence and bhattacharyya distance measures in signal selection. In *IEEE Transactions on Communication Technology*, volume 15, pages 52–60, 1967.

[214] C. Berge. Two theorems in graph theory. *Proceedings of the National Academy of Sciences of the United States of America*, 43(9):842–844, 1957.