Scalable and Adaptable Tracking of Humans in Multiple Camera Systems

A Gilbert

Submitted for the Degree of
Doctor of Philosophy
from the
University of Surrey

Centre for Vision, Speech and Signal Processing
Faculty of Engineering and Physical Sciences
University of Surrey
Guildford, Surrey GU2 7XH, U.K.

February 2008

Abstract

The aim of this thesis is to track objects on a network of cameras both within (*intra*) and across (*inter*) cameras. The algorithms must be adaptable to change and are learnt in a scalable approach. Uncalibrated cameras are used that are spatially separated, and therefore tracking must be able to cope with object occlusions, illuminations changes, and gaps between cameras.

The consistency of object descriptors is examined. In construction of robust appearance histogram descriptors, the histogram bin size, colour space and correlation measures are investigated. The consistency of object appearance is used as a measure of success for the possible solutions for tracking objects both intra and inter camera. The choice of descriptor will strongly affect tracking performance, hence these results are important and referred to throughout the thesis.

Crowded scenes of people would cause an appearance based individual tracker to fail. Therefore a novel solution to the problem of tracking people within crowded scenes is presented. The aim is to maintain individual object identity through a scene which contains complex interactions and heavy occlusions of people. The strengths of two separate methods are utilised; a global object search seeds positions to a localised frame by frame tracker to form short tracklets. The best path trajectory is found through all the resulting tracklets. The approach relies on a single camera with no ground plane calibration and learns the temporal relationship of objects detections for the scene. The development of a two part method allows robust person tracking through extensive occlusions and crowd interactions.

In addition to tracking objects within crowds, this thesis presents a number of contributions to the problem of tracking objects across cameras. A scalable and adaptable approach is used across the spatially separated, uncalibrated cameras with non overlapping fields of view (FOV). The novel approach fuses three cues of appearance, relative size and movement between cameras to learn the camera relationships. These relationships weight the observational likelihood to aid correlation of objects between cameras. Individually each cue has a low performance, but when fused together, a large boost in correlation accuracy is gained. Unlike previous work, a novel incremental learning technique is used, with the three cues learnt in parallel and then fused together to track objects across the spatially separated cameras. Incremental colour calibration is performed between the cameras through transformation matrices. Probabilistic modelling of an object's bounding box between cameras, introduces a shape cue based on objects relative size, while probabilistic links between learnt entry and exit areas on cameras provides the cue of inter camera movement. The approach requires no colour or environment calibration and does not use batch processing. It learns in an unsupervised manor and increases in accuracy as new evidence is accumulated overtime. Extensive testing is performed with 7 days of video footage using up to eight cameras with an hour of groundtruthed data. The use of these key developments allow for a flexible and adaptable approach to tracking people and objects intra and inter camera.

Key words: Human Tracking, Human Detection, Human Surveillance,Correlation Measures

Acknowledgements

I would like to thank Richard Bowden, for all his support and guidance throughout the last three years. Having such spirit and interest for computer vision research has helped me enjoy and pursue the work through the years.

I would like to thank the Faraday Partnership and Overview Ltd for the financial contributions that have made the research possible. In addition for providing a good insight into industry experiences.

Many people in the Centre for Vision , Speech and Signal Processing have taken time to provide advice and assistance and my thanks goes out to them.

Finally the constant support and encouragement of my wife Aseel and my family during this PhD has been invaluable.

Nomenclature

Symbol	Explanation
CAVIAR	Context Aware Vision using Image-based Active Recognition
CCCM	Consensus-Colour Conversion of Munsell Colour Space
CONDENSATION	Conditional Density Propagation
CLUT	Colour Lookup Table also known as CCCM
GMM	Gaussian Mixture Model
HI	Histogram Intersection
HSV	Hue, Saturation, Value Colour space
Inter Camera	Between multiple Cameras
Intra Camera	Within the one Camera
MCMC	Markov Chain Monte Carlo
MI	Mutual Information
FOV	Field of view
Real Time	Frame refresh rate of 25fps
RGB	Red Green Blue Colour Space
Source	Entry Point on a camera
Sink	Exit Point on a camera
SVM	Support Vector Machine

Symbols

Symbol	Explanation
x, y	Pixel location
Σ	Covariance
α	Gaussian Learning rate
k	Gaussian Distribution
K	Number of Gaussian Distributions
η	Gaussian probability density function
ω	Weight of Gaussian Distribution
μ	Mean
T	Time Reappearance Threshold
t	time
σ^2	Variance
r, s	Histograms containers
u, i, j	Histogram bins
m	Number of histogram bins
$f_i(r)$	Frequency Histogram of histogram r, bin i
Δi	Histogram Bin Width
δ	Delta Function
ρ	Bhattacharyya Coefficient
X	System Variables
A	State Transition Matrix
S	Covariance of Innovation
KG	Gain Matrix
n	Process Noise
z	Measurement including noise
l	Measurement
v	Measurement noise
ξ	Covariance of estimated system error
W	Number of head and shoulder detections
D	head and shoulder Detection
$\psi*$	Reference Appearance model of Object
ψ	Appearance Model of Object
S_t	Object State at time t
τ	Number of states in a frame
β	Region ID
O	Object
H	Transformation matrix

List of Publications

1. Gilbert A, Bowden R; Incremental Modelling of the Posterior Distribution of Objects for Inter and Intra Camera Tracking. *In Proc. British Machine Vision Conference* 2005.

2. Bowden R, Gilbert A, KaewTraKulPong P; Tracking Objects Across Uncalibrated Arbitrary Topology Camera Networks. *Chapter 6, In Intelligent Distributed Video Surveillance Systems* 2005.

3. Gilbert A, Bowden R; Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity. *In Proc European Conference Computer Vision* 2006.

4. Gilbert A, Bowden R; Multi Person Tracking within Crowded Scenes, *In Proc. Workshop on Human Motion, Understanding, Modelling, Capture and Animation, Human Motion Workshop in IEEE International Conference on Computer Vision*, 2007.

5. Gilbert A, Bowden R; Incremental, Scalable Tracking of Objects Inter Camera. *In Computer Vision and Image Understanding CVIU* Vol 111 Pages 43-58, 2008

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The motivation behind this work is to aid the operator's decision process by tracking objects accurately within multiple spatially separated cameras. The tracking occurs on (intra) and between (inter) cameras. The tracker should require no colour or spatial calibration about its environment. The techniques have no *a priori* data, but learn and are adaptable to the camera relationships over time. This will cause object correlation accuracy and interactions within the cameras to improve the tracking over time.

In this chapter, the applicability and use of object trackers, together with the goals of the research is presented. The social relevance of the work is given followed by the global structure and key contributions of this thesis based on scalable and adaptable tracking are discussed.

Automated tracking of objects in long range video is a large and growing field with many applications. It is used within many vision applications, these include

- tracking and identifying players within sports monitoring [70, 74, 112].
- tracking the movement of vehicles on roads [62].
- the construction of smart rooms or offices [45].

- the tracking of body parts for Human Computer interaction and perceptual user interfaces [37]
- The tracking of features within video sequences to allow for identification within a static database in automated video content retrieval [8]
- Use of the surveillance cameras in order to automatically track and follow people between cameras [13, 26]

Many of these approaches use few cameras or are in a restricted experimental environment. The proposals within this thesis aim to use an incremental and flexible approach; this allows a greater number of cameras to be used with a more flexible setup of equipment.

Tracking individual objects on surveillance cameras remains a difficult problem due to the complex interactions and occlusions that occur. Human tracking is a particularly difficult field as humans are deformable objects, meaning there is no fixed shape, size or colour that can be learnt. In addition, often the environments in which humans are tracked have very challenging conditions, such as illumination changes, background clutter, or occlusions from the background or objects themselves. To be able to achieve robust tracking there are a number of issues to consider which are examined through this thesis. The choice of features or descriptor used to represent the object is crucial. The ability to handle short and long term occlusion is also important, similarly, tolerance to the appearance variance of objects is necessary.

Surveillance cameras are increasingly being used as a tool to monitor and deter crime. As a result, there are large numbers of cameras which lack effective continuous monitoring due to the limitations of humans in managing large-scale systems. Therefore, tools to assist and aid the operator's decision process are essential. Visual surveillance systems are commonly placed in large areas with high levels of dense traffic such as in airports, rail stations and shopping centres.

This results in large numbers of cameras in constantly changing environments meaning that tracking must be adaptable to changes while scalable in design. This work proposes techniques to address these issues.

In Chapter 2, previous work related to tracking and the detection of objects is discussed. Significant methods from the past are presented, with their contributions and limitations highlighted. Chapter 3 provides a detailed examination of techniques used in later chapters.

This thesis consists of a number of key contributions over three core chapters / areas. Suitable descriptors to track objects are first examined in Chapter 4. A number of techniques to form appearance descriptor models are investigated. Different colour spaces and correlation methods are proposed, while quantisation is used to introduce illumination invariance. The use of quantisation allows for Parzen windowing to be employed to remove bin size constraints. Using groundtruthed data, the colour consistency of the techniques to form appearance descriptors is found for objects both inter and intra camera.

In Chapter 5, once the optimum combination of colour space, correlation and quantisation method are found, the results are employed . Within this chapter a novel approach is proposed to effectively track people on a single camera within a crowded scene with no ground plane information. The use of a head and shoulder detector provides “seed” object positions, these are tracked using a Mean Shift optimisation and terminated once no longer accurately tracking the original object. These short tracklets are combined with dynamic programming to produce a single trajectory of individuals through a sequence containing multiple occlusions and interactions.

Chapter 6 extends tracking into a network of up to eight cameras over two floors in real time (25fps). The system has no initial calibration or *a priori* information and contains cameras with both overlapping and non-overlapping field of view. The inter camera tracking is based around three individually weak cues of colour, time and shape. For scalability, the cues are learnt incrementally over time using observed correspondences that occur. The cues will learn in an unsupervised manner the relationships between the eight cameras, allowing tracking accuracy to increase. The scalable and adaptable real time algorithm is run for up to five days to show stability of accuracy. Extensive testing on three difference sequences evaluate inter and intra camera object tracking against groundtruthed data. The thesis is concluded in Chapter 7, examining the findings from each chapter, together with possible directions the work could be progressed in the future.

Chapter 2

Background

This thesis applies both tracking and detection theories to the security surveillance field. Relevant background work is therefore discussed in both the detection of objects and their tracking. The work within tracking is then further subdivided into single and multiple camera approaches both with and without overlapping fields of view between cameras.

2.1 Feature-based Object Detection

The detection system of Rowley *et al* [90] consisted of two neural networks, trained to detect frontal, upright faces in gray scale images. The first, faster network, performs an initial sweep to pre-screen candidate regions for the second, slower and more accurate detection. A similar idea, in the form of a cascaded detector was employed by Viola and Jones [107], who proposed a real-time face detection system. It used simple Haar-like features [64], with training and feature selection performed by Schapire *et al*'s AdaBoost [91]. They proposed the use of integral images to reduce computing costs, and extended the notion proposed by Rowley [90] through the use of a cascade of increasingly complex classifiers. The use of cascades allows for a low computational cost at run time. With high computation for the offline training of the classifier Oren *et al* [79] and Papageorgiou *et al* [82] used contour matching to train a detector to detect a full human body. However this method did not cope well with occlusions as a single contour around the body was used, and this was easily corrupted by occlusions. Mohon *et al* [71] proposed to solve this by sub-dividing the human body into its constituent parts of head, legs, left and right arms. These detected sub parts were grouped and classified by a Support Vector Machine (SVM), to determine actual person configurations [105]. Forsyth and Fleck [34] introduced body plans for assembling body parts. The body parts were simplistic pairs of parallel edges, these were then assembled by Ioffe and Forsyth [46] using projected classifiers. However, due to the simplistic features used, failure occurs in the presence of clutter or baggy clothing. Sigal *et al* [95] used a conditional probability distribution to model body part relations. However this was defined in 3D, requiring at least three stereo images which was a major constraint. Mikolajczyk *et al* [69] modelled humans as flexible combinations of boosted face, torso and leg detectors. Parts are represented by the co-occurrence of orientation features based on 1st and 2nd derivatives. The procedure is computationally expensive, but robust part detection is the key to the approach. Robust detection was possible, grouping different features into a single classifier. This is a “bag of words” approach, where each feature is a word and they are grouped together into a “sentence”. Micilotta *et al* [66] estimate the location and approximate 2D pose of humans through detection. They learnt individual body parts, and applied a coarse heuristic to eliminate outliers. An apriori mixture model of upper body configurations was then used to provide a pose likelihood for each configuration . The parts are then combined to form a joint-likelihood model. The combination of detectors allows for a more robust classifier allowing it to reject false matches.

2.2 Background Modelling

The detection of objects on individual images via a global search technique is a popular and fast moving field. However , if a video stream is available, it is sensible to use the history of the sequence in the current frame. One method, is to identify motion. Lucas and Kanade [63, 96, 55] proposed an image based correlation approach which is commonly used to compute optical flow. Optical flow computes a motion vector on a per pixel basis corresponding to the image velocity and is successively updated on a frame by frame basis. However, this is a computationally expensive method, and often the motion vectors of the pixels will have a high noise level, although this can be reduced by smoothing and sub sampling.

The process of background subtraction is a more popular method of separating the background (static parts) from the foreground regions (dynamic parts) of interest. The pixels that are changing colour are identified as foreground, allowing the static background to be ignored or removed. Extensive use of a dynamic background subtraction algorithm is made within this thesis.

Initial research into background segmentation resulted in the technique called chroma-keying, this was first used by Larry Butler, who won the Academy Award for Special Effects for the Thief of Baghdad in 1940. This was a hardware based technique using a screen behind the foreground object. The screen is of a constant colour, normally green or blue. The use of a constant colour allows the background screen pixel to be identified and therefore removed. A software approach was first presented by Smith and Blinn [97] and this is still a popular technique for constrained environments such as film or virtual reality. However, the need for specialist equipment limits its use to these constrained application domains.

Most software background segmentation can be classified by two main methods;

- Non-adaptive, for example frame differencing [89]. Non-adaptive background subtraction methods require manual re-initialization to acquire the static background image. Without initialization, errors in the background accumulate over time, making non-adaptive methods unsuitable for unsupervised, long-term tracking applications especially where illumination changes.

- Adaptive, these include the mean image over time, alpha blending [40], Kalman filtering [88], and Gaussian Mixture Models [100]. These approaches adapt to changes in the background over time.

Rosin and Ellis [89] made use of a simple technique known as frame differencing. Where successive images are subtracted from each other and thresholded to show pixels that have changed colour and therefore could contain motion. The problem with this technique is that segmentation fails when the foreground is a similar colour to the background and it can only be applied inside under controlled lighting as it is very sensitive to lighting or shadow changes. Haritaoglu *et al* [40] learn the background scene during a period of no foreground object by representing each pixel by three values; its minimum and maximum intensity values and the maximum intensity difference between consecutive frames observed during this training period.

Kalman filtering approaches such as that proposed by Ridder [88] can provide a partial solution. While Wren [110] with the *Pfinder* system proposed a per pixel background model, where each pixel had a mean colour value and a distribution centred at that mean, it was however, sensitive to initialisation inaccuracies. An extension of this is the use of multiple Gaussian distributions on a per pixel basis to model the individual pixel history. This approach was originally presented by Stauffer and Grimson [100]. This method uses mixtures for each pixel to provide a more detailed model of the background while maintaining a low computational cost. A more detailed explanation is given in Section 3.1. Despite its success, shadows can cause problems as they are incorrectly classified as foreground. Therefore, work has taken place to identify and remove the shadows.

Shadow Removal

Shadows can cause serious problems to segmentation by distorting the colour and shape of objects or giving false positive results. Cucchiara *et al* [25] proposed to use the HSV colour values to identify shadow areas, as they found that if a shadow is cast on a background pixel, the hue and saturation components change, but within a threshold if a shadow is present. Horprasert *et al* [41] labels shadows depending on the distortion of the brightness and the distortion of the chrominance of the difference.

While the Stauffer and Grimson method of using a mixture of Gaussians works well for most backgrounds, it can incorrectly label shadows as foreground. To remove this constraint, KaewTraKulPong and Bowden [16] proposed a technique based on Gaussian mixture models, to identify moving shadows that otherwise would be incorrectly labelled as foreground. To do this they make use of the brightness and chromaticity of pixels. Each non-background pixel is compared to the current background model, and if the difference in both chromaticity and brightness are within a threshold, the pixel is considered to be a shadow, and labelled as background.

An implementation of the approach by KaewTraKulPong and Bowden was available and this together with its real time computation meant it was chosen to be the segmentation method throughout this thesis. A more detailed explanation is given in Section 3.1. In Chapter 5 it is used to reduce the false positive rate of the head and shoulder detector and to improve the reliability of the tracker. In Chapter 6 its high speed and accuracy of detecting non-background objects in the videos allows for its use as the object detector for tracking objects cross cameras in real time.

2.3 Foreground Modelling

After background modelling has segmented foreground objects of interest, between frame tracking of the objects is possible. The way in which the tracked object is represented is crucial to the success of the tracking. When there is no predefined explicit shape model, some possibilities (as shown in Figure 2.1), are the bounding box 2.1(b), an ellipse [24] 2.1(a), the contours of a blob [47] 2.1(c), or the blob itself [14]. The bounding box shape is often referred to as the kernel. If there is an explicit shape model, a stick figure can be used, or every body part can have its own box [110]. When modelling the foreground in order to track areas of interest, the next distinction is the source of the image - single or multiple cameras, both with advantages and disadvantages.

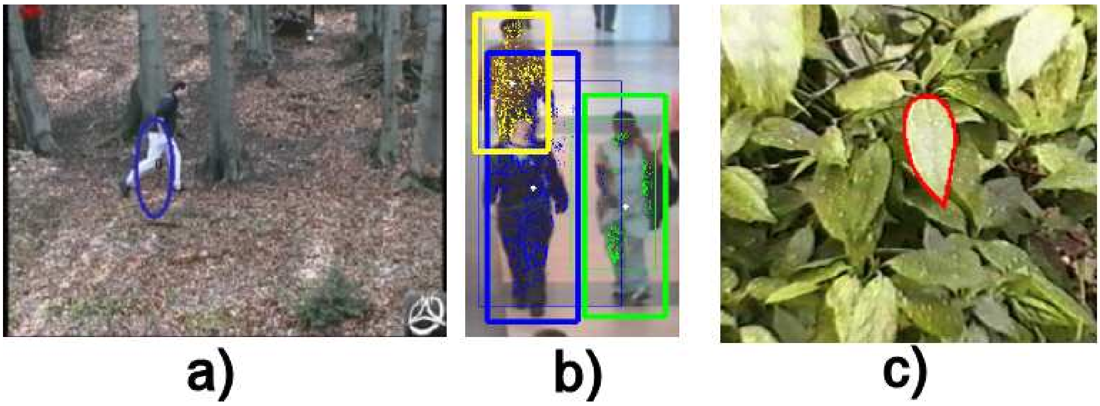


Figure 2.1: Examples of object representation when tracking, (a) uses an ellipse. (b) uses a bounding box. (c) uses the contour of the object

2.3.1 Single Camera tracking

Although very promising results have been presented through the use of multiple cameras, there are practical restrictions on having multiple cameras covering large scale installations due to cost. In addition, a single camera allows for simple and easy deployment. Therefore, a number of algorithms have been proposed to track objects on a single camera.

BraMBLe is a blob-based method proposed by Isard and MacCormick [48]. This is a multiple blob tracker that generates a blob-likelihood based on a background model and appearance model of the tracked objects. A Particle Filter framework is used to track an unknown number of people. Linking blobs together and learning their relationships, it is used by Bose *et al* [14] to track multiple interacting objects. Particle Filters were also used by Okuma *et al* [78] in conjunction with a boosted detector to help remove false particles in the filter to track fast moving ice hockey players.

The motion of objects can be used with a Kalman Filter [109] to track objects with an assumed constant velocity, as proposed by Xu *et al* [112] or Iwase and Saito [50]. Both authors use the motion model to track football players, while Boykov and Huttenlocher [17] use a similar model to track vehicles on highways. Koller *et al* [62] employ a contour tracker based on intensity and motion with a Kalman Filter for car surveillance.

Through the use of appearance, the accuracy of tracking can be significantly increased. Both Comaniciu *et al* [24] and Bradski [18] use Mean Shift to track regions based on correlation to a reference colour model. The search is deterministic using a metric derived from the Bhattacharyya coefficient as an appearance similarly measure; it proceeds iteratively from the final location in the previous frame so as to minimise the distance measure to the reference colour model. The advantages of this method is that no dynamic model is needed in advance. This copes with low resolution, deformable objects and partial occlusion, and can be extended with a Kalman Filter to use the velocity of the objects to improve tracking. However it will fail when parts of the background exhibit similar colours or when the tracked object is heavily occluded. Khan and Shah [58] use images segmented into different classes, that are then modelled by Gaussian mixtures. A conventional Bayesian-based tracking process is performed to track people. In the case of occlusions, the Gaussian models of visible objects are updated while the occluded objects are not. Senior [92] similarly learns a probabilistic model of the appearance of the tracked objects to track through occlusions and illumination changes. McKenna *et al* [65] use colour information to disambiguate occlusions that occur during tracking and to provide qualitative estimates of depth ordering and position during the object occlusion. Nillius *et al* [75] track individual football players until they merge, creating a new track identity until the player splits or merges with further people. This creates a “track graph” of the merging and splitting between players. Then the feature vectors of an individuals appearance is used to find the most likely set of paths for a given target based upon appearance.

In order to address non-Gaussian movement of objects, Isard and Blake [47] introduced CONDENSATION (Conditional Density Propagation). This is a factored sampling method introduced by Grenander [38] iterated over successive frames. In factored sampling, each sample in the sample set is weighted by a weight proportional to the sample probability at the previous time step. The resulting sample set will then represent the conditional probability. A similar sampling method was developed by Gordon *et al* [36] and Kitagawa [60] presented as Monte-Carlo methods. These iterative factored sampling algorithms are forms of Particle Filters. The basic idea of a particle filter is that random sampling is used to estimate a Bayesian, often multi-modal, model. CONDENSATION is used with active deformable contours to track a moving outline with substantial clutter. However this requires a large number of particles to accurately track the contour and is computational expensive if the dimensionality of the search space is high. Hue *et al* [43], Khan *et al* [59] and Vermaak *et al* [106] extended the particle filter framework to track multiple objects. Khan uses a MCMC (Markov Chain Monte Carlo) based particle filter and Vermaak uses a mixture particle filter for each tracked target. Okuma *et al* [78] combines the mixture particle filter with AdaBoost [107] to detect multiple people and track them in front of a cluttered background with a particle filter. Perez *et al* [85] and Nummiaro *et al* [77] proposed a particle filter making use of the colour histograms to track objects in a robust method. Giebel *et al* [35] used the multiple cues of shape, texture, and depth information from the image within a Particle Filter Bayesian framework, to track using learnt spatio-temporal object shapes in challenging scenes on single cameras.

Dealing with occlusion can be challenging with a single camera view. Zhao and Nevatia [115] rectified video frames to a predefined ground plane and modelled the targets in 3D space with a body shape model. The shape estimate allowed improved occlusion handling to estimate people in crowds. The tracking operated in a 2-frame interval, this caused some ambiguities due to the local view of trajectories. By examining the complete trajectory of objects some of these ambiguities can be resolved. More recently, Wu and Nevatia [111] used multiple body part detections to cope with minor body occlusions similar to the part-based object detection and recognition approach proposed by Mikolajczyk *et al* [69]. Wu combined the body parts with a Bayesian technique and tracked using Mean Shift, with an occlusion detection. This enables it to track multiple people through minor occlusions effectively on a single camera.

2.3.2 Multiple Overlapping Cameras

The use of multiple cameras which share a similar field of view was the focus of some of the earliest work into tracking people across cameras [56] [22]. They inherited many of the aspects of single camera systems, and often assume that the object is visible at all times in at least one camera. This provides a very robust platform for tracking moving objects, especially within crowds [33]. However, the requirement of having multiple cameras overlapping each other can be impractical, with many experiments using over five cameras covering a single room. This makes the techniques less suitable for real world deployment due to the large number of cameras required and physical constraints upon their placement.

The use of multiple, wide baseline cameras allows simple occlusion reasoning and through camera calibration a 3D environment can be built of the scene. This area has seen significant research and success. Early work by Kelly *et al* [56] required both camera calibration and overlapping fields of view. These were needed to compute the handover of tracked objects between the cameras. Additionally, Chang [21] created a 3D model of the environment using epipolar geometry, to allow for the registration of objects across the different overlapping cameras.

Modelling the motion of tracked object during times of occlusion has been achieved using Kalman filters. Mikic *et al* [67] found 3D points from projections of points belonging to binary blobs, and applied a Kalman filter to track objects. Black *et al* [13] also used a Kalman filter to estimate the trajectory projection during occlusion, to simultaneously track in 2D and 3D. Dockstader and Tekalp [28], Chang [22] overcame occlusion in multiple object tracking by fusing the information from multiple cameras. Trivedi *et al* [104] used a Kalman filter to model the motion of the tracked target and to determine the optimum camera for tracking in a multi-camera system.

Orwell *et al* [81] present a tracking algorithm to track objects using appearance.

They model the connected blobs obtained from background subtraction using a colour histogram and then match and track multiple objects. In [80], Orwell *et al* present a multi-agent framework for determining whether different agents are assigned to the same object seen from different cameras. This method would have problems in the case of partial occlusion where a connected foreground region does not correspond to one object, but has parts from several of them. Cai and Aggarwal [20] compute correspondences between multiple cameras to extend the single view tracking of people. They use a background segmentation across calibrated video streams to extract human shape. Feature points are then extracted and tracked in a single view. The system then switches to another view when the current camera no longer provides a good view of the person's features. Occlusion and overlapping people will cause blob based segmentation to fail. Therefore Figueoa *et al* [32] split incorrectly formed blobs based upon their pixel appearance. More recently, Morariu and Camps [72] proposed a method based on dimensional reduction to learn the correspondence between the appearance of people across multiple views. This uses the appearance of the person in one camera to compensate for occlusion of the same person in another view.

Object appearance can be applied as a strong cue in overlapping cameras to maintain object identity. Colour based appearance and motion information is used by Kang *et al* [54]. They leart the limits of the field of view (FOV) for overlapping cameras to maximise a joint probability of the 2D and 3D position of individuals. When a person is visible in one camera, other cameras where the person *should* also be visible are examined. Mittal and Davis [4] use observed intensity models of people to segment images in up to 6 synchronised overlapping cameras. Regions are matched across views using a region-matching stereo algorithm yielding 3D points potentially lying inside objects. These points are projected onto the ground plane to form an object location likelihood map. Nummiaro *et al* [76] use appearance matching from multiple cameras to determine the optimum view for tracking an individual. They use multiple models for the target to ensure robust tracking even when the object appearance changes considerably through pose changes.

The use of probabilistic occupancy maps has seen some of the most recent and promising progress. Through the use of a discretised ground plane map of the cameras field of view, occluded objects are tracked through reasoning. Beymer [11] detects areas of disparities on stereo images to produce a score map of the likely location of individuals. A Kalman filter is combined with this to model the motion of objects. A Visual Hull approach can be used to compute the occupancy map as proposed by Yang *et al* [114]. Fleuret *et al* [33] use the information from multiple cameras to produce a probabilistic occupancy map based on the dynamics and appearance models of the objects. Dynamic programming is then used to track the multiple objects through significant occlusion and lighting changes.

2.3.3 Multiple Non-Overlapping Cameras

The use of non-overlapping cameras creates very different problems to overlapping cameras. The handover of objects between cameras cannot be explicitly observed and therefore reasoning must be used. However, they are far more flexible in setup as they do not require overlapping FOV and the physical constraints that must be applied to satisfy overlapping FOVs. Therefore these techniques have the greatest real world potential application.

Calibrating the cameras provides increased accuracy and often no additional learning once the calibration is complete. Although there are disadvantages to this approach: for example, once the calibration has been completed, should the system change, a recalibration would be required. In addition, the time to perform the calibration could increase exponentially as the number of cameras increase. Kettnaker and Zabih [57] presented a Bayesian solution to tracking people across cameras with non-overlapping FOVs. The technique required calibration, with the user providing a set of inter camera transition probabilities and their expected duration *a priori*. This means that the environment and the way people move within it would need to be initially measured. This would be a large and exponentially increasing constraint as the number of cameras used increased. Chilgunde *et al* [23] track objects across blind regions inter camera using a Kalman filter. However it is assumed that the ground plane is known between the cameras.

Probabilistic or statistical methods have recently seen some of the greatest focus in solving inter camera tracking. They all use the underlying principle that through accumulating evidence of movement patterns over time, it is likely that common activities will be discovered. Huang and Russell [42] presented a probabilistic approach to tracking cars on a highway, modelling the colour appearance and transition times as Gaussian distributions. This approach is very application specific, using only two calibrated cameras, with vehicles moving in one direction in a single lane. Javed *et al* [51] present a more general system by learning the camera topology and path probabilities of objects using Parzen windows. This is a supervised learning technique where transition probabilities are learnt during training using a small number of manually labelled trajectories. Dick and Brooks [26] use a stochastic transition matrix to describe patterns of motion both intra and inter camera. The system required an offline training period where a marker was carried around the environment. This would be infeasible for large systems and can not adapt to cameras being removed or added *ad hoc* without recalibration. For both systems [51, 26], the correspondence between cameras has to be supplied as calibration data *a priori*. Pasula *et al* [84] uses a Markov Chain Monte Carlo (MCMC) method to identify objects across a multi-camera network. The MCMC allows for an accurate estimation of the orgin/destination transition times even when individual links in the sensor chain are unreliable.

Ivanox*et al* [49] hand-coded source and sink models (entry and exit areas on cameras), for use in high level event reasoning. Thus allowing them to differentiate a person who entered the scene walking verses a person in a car. Shet *et al* [94] also hard-code logic into their system to allow it to reason about the identity of people within a video system. The use of manually labelling information can provide accurate results, however it is impractical for a larger design.

Stauffer [99] finds the entry and exit points of objects within a camera network using common paths. KaewTraKulPong and Bowden [52] or Ellis *et al* [30] do not require *a priori* correspondences to be explicitly stated; instead they use the observed motion over time to establish reappearance periods. Ellis learns the links between cameras, using a large number of observed objects to form reappearance period histograms between the cameras. KaewTraKulPong uses appearance matching to build up fuzzy histograms of the reappearance period between cameras. This allows a spatio-temporal reappearance period to be modelled. In both cases batch processing was performed on the data which limits their application to the real world.

Colour is often used in the matching process. Black *et al* [12] use a non-uniform quantisation of the HSV colour space to improve illumination invariance, while retaining colour detail. KaewTraKulPong and Bowden [52] uses a Consensus-Colour Conversion of Munsell colour space (CCCM) as proposed by Sturges *et al* [102]. This is a coarse quantisation based on human perception and provides consistent colour representation inter-camera without explicit colour calibration. Most multi camera surveillance systems assume a common camera colour response. However, even cameras of the same type will exhibit differences which can cause significant colour errors. Pre-calibration of the cameras is normally performed with respect to a single known object, such as the GretagMacbeth [98] ColorChecker$^{\mathrm{TM}}$chart with twenty four primary colours used by Ilie and Welch [44]. Porikli [86] proposes a distance metric and model function to evaluate the inter camera colour response. It is based on a correlation matrix computed from three 1-D quantised RGB colour histograms and a model function obtained from the minimum cost path traced within the correlation matrix. Joshi [73] similarly proposes a RGB to RGB transform between images. By using a 3x3 matrix, inter channel effects can be modelled between the red, green, and blue components. More recently, Annesley and Orwell [7] model colour variation between cameras to enforce colour consistency between cameras, using the grey-world assumption to model the colour variation. This worked well, however, it was tested within a relatively restricted environment; the work within this thesis removes the need for some of those restrictions.

Chapter 3

Methods

This chapter will explain and describe some of the main computer vision methods used in the later work. There are three, main, low level techniques, used to identify the foreground, and track foreground objects; an Adaptive Gaussian Mixture Model background segmentation, Mean Shift and Kalman Filters. each is now examined in turn.

3.1 Adaptive Background Segmentation

The background segmentation technique used within this thesis is based on that originally presented by Stauffer and Grimson [100] and extended by KaewTraKulPong and Bowden [16]. The algorithm classifies on a per pixel basis, whether the pixel belongs to the background model it has formed for each pixel point. If it does not fit the model, the pixel is classified as foreground. The Stauffer and Grimson algorithm relies on assumptions that the background is visible more frequently than any foreground and that it has modes with relatively narrow variance. These assumptions are consistent with scenes in which the background clutter is generated by more than one surface appearing in the pixel view. The foreground vs background pixel segmentation is formed using a Gaussian mixture model on a per pixel basis, so for every pixel there are K Gaussian distributions. Each Gaussian distribution has a mean μ, a standard deviation σ and a weight ω. A large value of K provides more robust segmentation, but at the cost of slow system performance. Three to five Gaussians have been found to provide sufficiently robust segmentation, while still maintaining real-time performance.

At a given time t, the Gaussian probability density function, η for the kth Gaussian distribution, with a mean $\mu_{k,t}$, and covariance $\Sigma_{k,t}$ is given as

$$\eta(x_t, \mu_{k,t}, \Sigma_{k,t}) = \frac{1}{\sqrt{2\pi|\Sigma_{k,t}|}} e^{-\frac{1}{2}(x_t-\mu_{k,t})^T \Sigma_{k,t}^{-1}(x_t-\mu_{k,t})} \tag{3.1}$$

Therefore the overall likelihood that a pixel x_t fits this Gaussian is

$$P(x_t) = \sum_{k=1}^{K} \omega_{k,t} \times \eta(x_t, \mu_{k,t}, \Sigma_{k,t}) \tag{3.2}$$

where $\omega_{k,t}$ is the weight of the distribution, initialised as $\frac{1}{K}$. The weight on each distribution represents the probability that a colour of the image pixel remains the same, ie is part of the background.

An online approximation to expectation maximisation is then used to update the pixel distribution. For a given pixel intensity value x_t, it is compared to the existing distribution model components. A correlation is found if the intensity is within 3 standard deviations of any Gaussian component. If no correlation is found to the existing models, the Gaussian distribution with the lowest weight is replaced by a new one with the value of the pixel as the mean, with a initially high standard deviation σ_0 and a low weight ω_0. The matched distributions are then updated calculating a new mean μ and variance σ^2. The mean moves in the direction of the pixel value and is weighted by the learning rate α.

$$\mu_{k,t} = (1-\alpha)\mu_{k,t-1} + \alpha x_t \tag{3.3}$$

$$\sigma^2_{k,t} = (1-\alpha)\sigma^2_{k,t} + \alpha x_t \tag{3.4}$$

The weight of each distribution is updated: the correlated distribution has its weight increased, and the rest reduced. The learning rate, α, controls the amount with which the weights are updated. Equation 3.5 is the updated weight for the correlated distribution

$$\omega_{k,t} = (1-\alpha)\omega_{k,t-1} + \alpha \tag{3.5}$$

while Equation 3.6 shows the updated weight to the remaining K distributions

$$\omega_{k,t} = (1-\alpha)\omega_{k,t-1} \tag{3.6}$$

Next a decision is made as to whether the new pixel value is foreground or background. This is achieved using the equation 3.2 to compute the probability that a pixel intensity is background. In order to reduce computation all the distributions are sorted by ω. The distribution that is the most probable background model, and therefore the most correlated over time, will be matched first and therefore removes the need to match to the other distributions. Because there are only two parameters, the learning rate α, and the number of background models K, this algorithm is very easy to adapt to the environment. The parameter α, adjusts the speed at which static foreground becomes background and is tuned manually.

As there are multiple distributions, every pixel can be represented by one or more background colours. In this way, a repetitive dynamically changing background such as leaves on a tree, can still be classified as background. This will create a number of probable background models to correlate with the moving background object. The use of multiple distributions can also remove false positive foreground detections. These can occur when stationary foreground objects move, causing a false detection on the pixels previously occupied causing a *ghost* segmentation. Using multiple distributions, updating the background does not destroy the existing background colour model, which will move to a lower weighted distribution. Thus, if the object moves, the distribution describing the previous background still exists (with a lower weighting), and will quickly regain dominance in the model.

3.2 Mean Shift

Mean Shift is an optimisation technique used for many applications but popular as an appearance based tracker. The "Mean Shift" is the rate at which the target's kernel moves over succesive frames. A kernel is an bounding box covering the area of interest, in this case a person. The Mean Shift is computed by performing an iterated localised gradient accent from the target object's location in the previous frame. The normalised colour histogram appearance model r of a kernel of a new object to be tracked is found. To compute distances between the histograms, the Bhattacharyya distance measure is used. This measure was found to provide the most consistent intra camera colour correlation in tests in Chapter 4 (Figure 4.8 in Section 4.4.2) and by others [24]. In each new frame, the target moves toward its most probable position. To do this an algorithm is used that maximizes the Bhattacharyya coefficient

Bhattacharyya coefficient Maximisation

1. First, for candidate location y_0, the current candidate histogram $s(y_0)$ will be computed. After that the Bhattacharyya coefficient between this histogram and the model histogram r is evaluated

$$\rho[s(y_0), r] = \sum_{u=1}^{m} \sqrt{s_u(y_0) r_u} \tag{3.7}$$

2. For every pixel a weight is derived

$$w_i = \sum_{u=1}^{m} \delta[b(x_i) - u] \sqrt{\frac{r_u}{s_u(y_0)}} \tag{3.8}$$

where $b(x_i)$ is the bin for the colour of the pixel x_i, u is the current bin and δ is the Kronecker delta function, which is only true if both arguments are true. This means that every weight is the square root of the value of the model bin of the pixel colour, divided by the value of the candidate bin (ignoring empty bins).

3. To derive the new estimated location y_1, the *Mean Shift* is computed as

$$y_1 = \frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i} \tag{3.9}$$

then $s_u(y_1)$ can be updated and a Bhattacharyya coefficient between the new candidate histogram $s(y_1)$ and the model histogram r evaluated

$$\rho[s(y_1), r] = \sum_{u=1}^{m} \sqrt{s_u(y_1) r_u} \tag{3.10}$$

4. While $\rho[s(y_1), r] < \rho[s(y_0), r]$, the target has not yet been reach so the location of y_1 must be updated $y_1 \leftarrow \frac{1}{2}(y_0 + y_1)$

5. If $\|y_1 - y_0\| < \varepsilon$, the iterations stop, and the target has been found, otherwise the iteration is restarted at step 1 with a new candidate position: $y_0 \leftarrow y_1$.

The use of a local search area with iterations allows the Mean Shift technique to have low computational complexity, while maintaining a good approximation to the optimal trajectory of the target object's model.

3.3 Kalman Filter

The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) method to estimate the state of a process, in a way that minimizes the mean of the squared error. It can be used to estimate the state position of an tracked object at the current time interval based on measurements of the object location in previous time intervals.

To estimate the state of some variables X in a system, it is assumed that the system variables are controlled by equation 3.11

$$X_{t+1} = AX_t + n_t \tag{3.11}$$

where n_t is random process noise, and A is the state transition matrix, where t is the time between the time intervals t and $t+1$. Equation 3.11 can be expanded into a four state vector to track an object centroid independently in x and y in an image

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ dx_{t+1} \\ dy_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ dx_t \\ dy_t \end{bmatrix} + \begin{bmatrix} n(x) \\ n(y) \\ n(dx) \\ n(dy) \end{bmatrix} \tag{3.12}$$

The measurement z of the x, y location of the object can be represented by sum of the measurement, l, and measurement noise, v, as shown in equation 3.13

$$z_t = l_t + v_t \tag{3.13}$$

The Kalman Filter update and estimation process is be made up of four equations, where following two statistical conditions are true.

- On average, over time the estimate of the state must equal the true state value.
- The mean squared error on the estimated state must be minimised.

In order to estimate the next state, first the covariance of innovation S and gain matrix K are computed in equations 3.14 and 3.15

$$S_t = \xi_t + V \tag{3.14}$$

$$K_t = AP_tS_t^{-1} \tag{3.15}$$

Where the measurement noise is white Gaussian noise with a covariance V, and ξ is a covariance of the system estimate error. Then in order to estimate the next state, the prediction error and state estimate is updated in equations 3.16 and 3.17.

$$P_{t+1} = AP_tA^T + Q - AP_tS_t^{-1}P_tA^T \tag{3.16}$$

$$x_{t+1} = Ax_t + K_t(z_t - Ax_t) \tag{3.17}$$

Q is the covariance matrix of the process noise, while equation 3.17 computes the new estimated state, it consists of the state at the previous time interval x_t with a difference between the actual measurement z_t and previous estimated state weighted by the gain matrix. These four equations can then be repeated with $t = t + 1$, to continue predicting and updating the state of the tracked object.

Chapter 4

Consistency of Object Descriptors

This chapter investigates methods to measure the consistency of object appearance for tracking intra and inter camera. For a descriptor to be robust, it must maintain a high degree of consistency through object illumination and pose changes within and across cameras.

Appearance is one of the possible cues used to recognise and correlate objects. It is used in many computer vision tasks including image retrieval, 3D modelling and tracking. This chapter will explore three factors to consider when using an appearance based descriptor:

1. The selection of a colour space to represent the pixel intensities,
2. An optimal descriptor quantisation,
3. A correlation measure to discern between descriptors.

The consistency of an object's appearance is used to assess performance. The object will have a constant appearance between frames and cameras, however the illumination and pose changes will affect its appearance descriptor. Therefore the examination of the consistency of an object's appearance is used to distinguish between the three factors listed above. This will provide a measure of a technique's success as an appearance descriptor for an object.

4.1 Object Descriptor

An object descriptor is a container to represent object metadata used later for identification and correlation. In this chapter the descriptor will be based only on the object's colour appearance. Colour will be used as this provides enhanced discriminative detail over grey-scale. In addition, shape and motion are important descriptors to distinguish between multiple classes of objects. However, as this work is concerned with a single class of object i.e, humans, shape, or motion alone would not provide sufficient discriminatory information.

4.1.1 Localisation of Foreground Objects

To locate an object of interest, Connected Component Analysis is applied to the foreground pixels formed using the Gaussian Mixture Model method outlined previously in section 3.1. Connected Component Analysis scans the background binary mask and groups its pixels into components or blobs based on pixel connectivity. Once all blobs have been determined, the mean and variance of the connected pixels within the blob is used to produce a rectangular bounding box kernel centred on each object. A minimum blob size is used as a filter to remove the small areas of foreground noise and leave only the moving objects of interest with a rectangular kernel as shown in Figure 4.1. To efficiently describe the appearance of the discrete foreground object, a frequency histogram of the pixel intensity values is used.

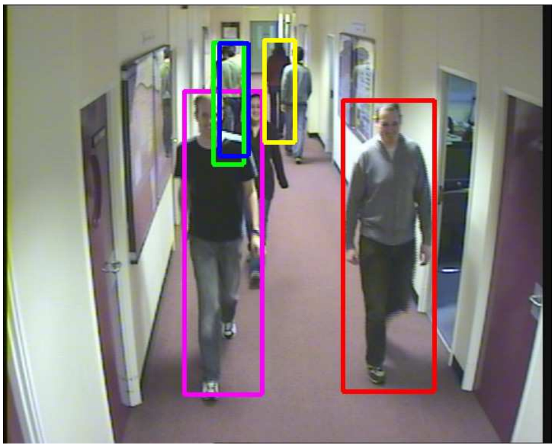


Figure 4.1: Objects detected by Connected Component Analysis on a image frame.

4.1.2 Colour Histograms

A measure of similarity to correlate between colours histograms called *Histogram Intersection* was first proposed for colour image retrieval by Swain and Ballard [103]. It is a method to model the appearance of an object as a discrete frequency probability density function. The use of colour within recognition is a critical perceptual cue. The appearance frequency distribution of an object is robust to a number of effects including rotation and perspective distortion. The histogram will vary slowly to changes in viewing angle, scale, and pose and occlusion of the person [103]. Through quantisation, a degree of invariance to illumination changes can also be introduced. This is especially important for cross camera tracking where image illumination has large variations as illustrated by Figure 4.2, which is a picture from an indoor set of cameras.

A colour space is defined by a number of axes (often three), the colour histogram is obtained by discretising the pixel intensities of the object's foreground bounding kernel and counting the frequency of each discrete colour that occurs in the area. Thus, the colours in an image are mapped into a discrete colour space containing m colours. A colour histogram of object I is an n-dimensional vector,

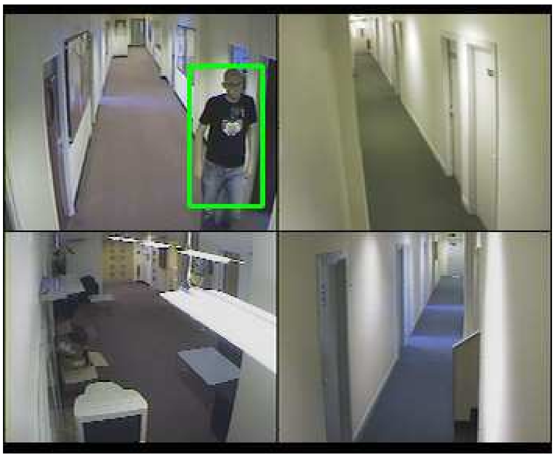


Figure 4.2: Examples of illumination variation between four close cameras all with white walls.

where each element represents the frequency of colour b in object I. The colour histogram model is given as $r_I = (u_1, u_2, ..., u_m)$. The histogram dimension, m is determined by the degree of quantisation for a given colour space. The standard RGB colour space of three axes may be quantized into r, g, and b bins for each axis. The histogram can then be represented as an n-dimensional vector whose length is given by the product, $m = r * g * b$.

4.1.3 Quantisation

The aim of quantisation is to reduce the storage requirements and sensitivity of the data. The ability of the histogram to model the object's appearance is affected by the width of the histogram bins. The bin width, Δi chosen for the objects appearance histogram is crucial to the system performance. If only a small set of samples is available or a large Δi is used, the model tends to converge to a singular solution. To remove this, the bin size, Δi can be reduced. This then leads to another question, how small the bin width should be to represent the colour distribution of the object. Too small a bin size can result in no discernible difference between objects. Therefore a measure based on the number of samples used to populate the histogram was used to determine the bin size. The relationships between bin size and sample established by Hadjidemetriou *et al* [39] was used. They recommended the number of bins $m_i = \frac{\beta}{\Delta i}$ where β is the total number of possible pixel values (for unquantised RGB 256*256*256), be proportional to the cube root of the number of samples, N_x.

$$\Delta i \propto \sqrt[3]{m_x} \tag{4.1}$$

In the following, different widths of Δi are examined to allow a constant to be chosen for the later work.

To further reduce the constraints on varying the bin sizes, two further solutions are possible. One is to use a different colour space that utilises a colour transformation obtained from consensus colours in the Munsell colour space, into eleven discrete fixed *primary* colours. This is explored in section 4.2. The other is to use a window function to reduce the dependency on the correct histogram bin size during quantisation, i.e. a Parzen window.

4.1.4 Parzen Windowing

Kernel density estimation (or Parzen windowing, named after Emanuel Parzen [83]) is a way of estimating the probability density function of a random variable. It superimposes kernel functions on each observation similar to convolution of a kernel with a observation signal. This enables the quantised histogram to be more robust to the choice of Δi as data is spread to multiple bins to minimise under complete population. The choice of the kernel used is important since it is conditioning the quality of the estimate. The most popular function is the gaussian distribution. It has a zero mean and variance σ^2, the choice of σ^2 is then very important as well, as this will determine the spread of the observation and is based on the number of available observations. Dowson and Bowden [29] named two types of Parzen Windowing, In-Parzen Windowing and Post-Parzen Windowing.

In-Parzen Windowing is designed to reduce the effect of different quantisation levels by convolving *each* observation u_i with the smoothing kernel prior to quantisation then populating multiple bins in the colour histogram. The In-Parzen window estimate function $P(x)$ is shown in equation 4.2

$$P(x) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{\sigma_n}\eta(\frac{u - u_i}{\sigma_n}) \tag{4.2}$$

where σ is the bandwidth or standard deviation of the window and is based on the number of observations or samples m. $G(x)$ is the unimodal kernel window function such that

$$\int_{\infty}^{-\infty} \eta(x)\ dx = 1 \tag{4.3}$$

For this work a 1D Gaussian PDF is used as the kernel, when applying this to equation 4.2, the In-Parzen window estimate function with a Gaussian kernel becomes

$$P(x) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{\sigma\sqrt{2\pi}}\ exp(-\frac{1}{2}(\frac{u - u_i}{\sigma})^2) \tag{4.4}$$

Performing In-Parzen Windowing during the histogram construction is comparatively expensive as it is dependant on the number of observations m used to form the histogram, and some loss of information still occurs. However In-Parzen Windowing is able to reduce much of the problem of over and under fitting data with an incorrect Δi, through the blurring of the bins around the actual observation. To reduce the computational costs of In-Parzen windowing, Post-Parzen Windowing involves superimposing a kernel with the observations, after *all* the observations have been sampled. Again the kernel used is often a Gaussian, however convolving with the kernel after all observations have been sampled, the computational cost is not dependant on the number of observations N_x, but the number of bins. However, it results in a loss of information due to the application of the gaussian after the quantisation has taken place, and it is not always smooth. Despite these constraints it is still able to be reduce the dependency of bin size that quantisation introduces.

4.2 Object Colour Space

A colour space can be defined as a model representing different frequencies of light in terms of intensity values. An object should have a constant appearance, however the illumination on the object will be constantly changing due to shadows or lighting. These changes will be detected by a camera and, if not dealt with, will make tracking or correlating the object appearance impossible. Normalisation or quantisation of the colour space can provide illumination invariance. For this work quantisation was used as it also is able to reduce the dimensionality of the data. Quantisation reduces the total number of unique colours, meaning that similar colours are labelled as the same. Several colour spaces and quantisation levels were investigated including the HSV quantisation (8x8x4) approach proposed by Black *et al.* [12], the Consensus-Colour Conversion of Munsell colour space, a colour Lookup table (CLUT) [15] and differing levels of traditional RGB quantisation. These colour space models are only a few of the many other colour space models available including YUV. They are just some of the more popular ones.

The HSV colour Model

The HSV (Hue, Saturation, Value) model, defines a colour space in terms of three constituent components:

- Hue: the colour type (such as red, blue, or yellow). Ranges from 0 → 360. Each value corresponds to a colour. For example: 0 indicates the colour red, 120 indicates green, and 240 indicates blue. The primary and secondary colours red, yellow, green, cyan, blue, and magenta occur at the vertices of the hexagons.

- Saturation: this is the intensity of the colour. Ranges from 0 → 1. Where 0 means no colour, 1 means intense saturated bright colour.

- Value: this describes the brightness of the colour. Ranges from 0 → 1, 0 is always black. Depending on the saturation, 1 may be white or a colour at its brightest.

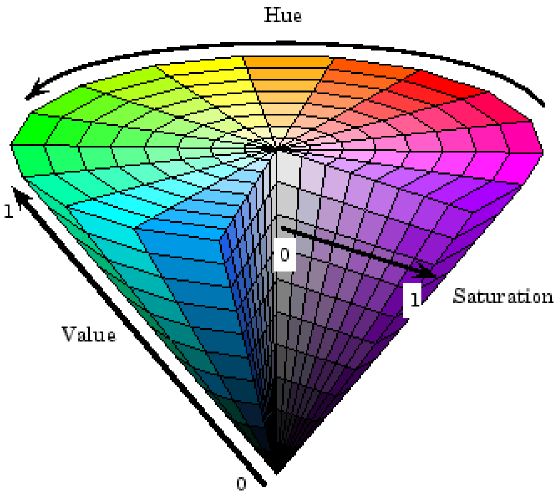


Figure 4.3: The HSV colour space represented by a hexagonal cone [2].

Figure 4.3 shows a visualisation of the HSV model as a cone. In this representation, the hue is depicted as a three-dimensional conical formation of the colour wheel. The saturation is represented by the distance from the centre of a circular cross-section of the cone, and the value is the distance from the pointed end of the cone. This colour space is often used within the computer vision as it is possible to separate the Hue and Saturation components from the Value component. As the Value component is the most affected by illumination changes, the overall histogram can be made more invariant to the illumination changes by increasing the bin size in this channel. While maintaining discriminatory information about the object's colour in the Hue and Saturation components. A Parzen window estimation could be applied to the HSV histogram, however it could not be a regular lattice, due to the conical shape of the colour space. Therefore to correctly estimate the HSV colour space with Parzen windows a conical gaussian would need to be applied. While within the RGB colour space a standard gaussian window kernel can be applied with ease. Therefore the Parzen window functions are only applied to the RGB colour space.

Conversion of Munsell Colour Space

The Colour Lookup table (CLUT) colour Space is a manually defined colour space. It utilises a colour transformation obtained from consensus colours in Munsell colour space [10]. Sturges *et al* [102] identified consensus areas within the colour space, where groups of colours were consistently labelled the same each time they were viewed by humans in a physiological study. Figure 4.4 shows the areas of colour. This meant it was possible to separate the colour space into eleven primary discrete colours, which have no overlap and have no other colour present. Looking at Figure 4.4 it can be seen that they are not linear or uniform in size or distribution. This means that Parzen windowing cannot be applied to the data to smooth the quantisation bins. Pixel values not occurring within a colour label are matched to the closest label using the Mahalanobis distance. The very coarse quantisation can provide good illumination invariance for object recognition both intra and inter camera, however it is possible that too much discriminatory information has been discarded. This could result in correlation failures between similar objects.

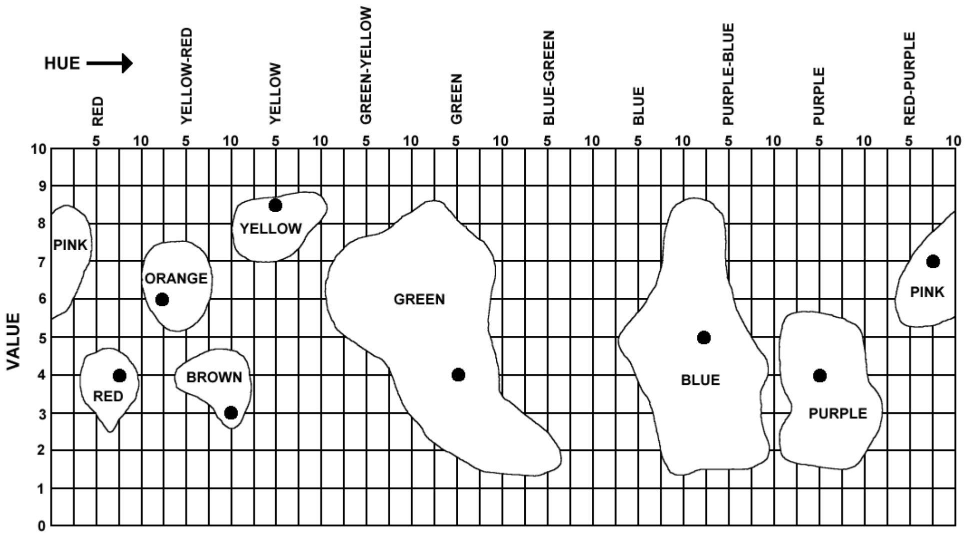


Figure 4.4: Location of consensus samples and focal colours on a two dimensional representation of the Munsell space as identified by Sturges *et al* [102].

The RGB Colour Model

The RGB (Red, Green, and Blue) colour space model is widely used in machine vision and in many digital-imaging devices. It is an additive colour model, because it describes what kind of light needs to be emitted to produce a given colour. Light is added together to create colour ranging from black to white. This colour space is specified by the chromaticities of its primary colours and its white point. The RGB system provides fast and simple computation, but it is neither perceptually uniform nor an intuitive colour space. However it is a linear model allowing for the use of Parzen windowing to help reduce its dependance on the quantisation bin size during histograms.

4.3 Similarity Measures

The correlation of objects both inter (between) and intra (within) camera requires the use of a similarity measure. Four possible techniques to compare the similarity of histograms were examined. Given two objects R and S to be correlated, a pair of normalised appearance frequency histograms will be produced from their foreground appearance within the kernel bounding box taken from the background mask. These can be represented as $f(R) = R(i)_{i=1...m}$ and $f(S) = S(i)_{i=1...m}$, each having m bins.

4.3.1 Histogram Intersection

Swain and Ballard [103] introduced a histogram matching method called Histogram Intersection. Given a pair of normalised histograms, $f(R)$ and $f(S)$ of objects I and J. Both histograms consist of m bins and the i-th bin $i = 1, ..., m$, is denoted with $f_i(I)$ and $f_i(J)$ respectively, the histogram intersection of the normalised histograms is as follows:

$$\rho_{HI}[f(R), f(S)] = \sum_{i=1}^{N} min(f_i(R), f_i(S)) \tag{4.5}$$

For two objects, the larger the value of the Histogram Intersection, the more similar the object pair is deemed to be. The technique is easy to use and has low computational complexity.

4.3.2 Bhattacharyya Coefficient

The Bhattacharyya coefficient [5, 53] is a popular correlation to compute the similarity of two probability distributions. It measures the separability of the two classes. The Bhattacharyya coefficient is closely related to the Bayes error [6]. Calculating the Bhattacharyya coefficient (derived from the Bayes error) involves integration of the overlap of the two samples.

$$\rho(R, S) = \int \sqrt{R(i)S(i)}\ \ di \tag{4.6}$$

where R and S is the colour distribution of the two objects to be correlated. For discrete probability distributions $f(R)$ and $f(S)$, equation 4.6 can be approximated as the integral of the scalar product of the two vectors (one for R and one for S), defined as:

$$\rho[f(R), f(S)] = \sum_{i=1}^{m} \sqrt{f_i(R) f_i(S)} \tag{4.7}$$

The coefficient interval is $[0, 1]$, where a value of 1 is complete correlation, or 0 if there is no correlation at all due to the multiplication by zero in every bin. For experimental work, due to the discretisation of the continuous probability density functions into histograms, zeroes do occur, and are replaced by a small value, 0.0001 due to the use of the square root function.

4.3.3 Mutual Information

The Mutual Information measure [101, 61] is based on the shared information of the overlapping part of two object's appearance histograms. This information is obtained using Shannon entropy [93], known as the measure of uncertainty. It is used as a similarity metric to measure the mutual dependence between the two appearance histograms. It uses the joint appearance frequency histogram between images in addition to the separate appearance histograms. This allows basic spatial information to be encoded into the metric, while maintaining the speed and invariance properties of separate histograms.

Given a pair of objects I and J respectively, the Mutual Information MI is the sum of their entropies H minus their joint entropy.

$$MI[f(R), f(S)] = H_R + H_S - H_{RS} \tag{4.8}$$

Where H_R is object R's Entropy. Entropy is a measure of the average uncertainty associated with a random variable. It is described by the object appearance intensity i within the kernel box, and the number of pixels in a frequency histogram bin f_k.

$$H_R = -\sum_{i=1}^{m} \frac{f_i(R)}{\sum f_i(R)} \ln \frac{f_i(R)}{\sum f_i(R)} \tag{4.9}$$

While the joint entropy H_{RS} of both objects pixel intensities within their kernel bounding boxes can be given by;

$$H_{RS} = \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{f_{ij}(RS)}{\sum f_{ij}(RS)} \ln \frac{f_{ij}(RS)}{\sum f_{ij}(RS)} \tag{4.10}$$

where $f_{ij}(RS)$ is the joint probability between the two objects R and S. If objects R and S are found to be correlated H_{RS} will be smaller than $H_R + H_S$.

4.3.4 Chi-square distribution

The chi-square distribution (also chi-squared or χ^2 distribution) is widely used in statistical significance tests. It is used to estimate how closely an observed distribution or histogram matches an expected distribution, this is often called the *goodness-of-fit* test. Chi-square is calculated by finding the difference between each object's I and J colour distribution in each histogram bin, squaring them, dividing each by the object's J histogram bin frequency, and taking the sum of the results.

$$\chi^2 = \sum_{i=1}^{N} \frac{(f_i(R) - f_i(S))^2}{f_i(S)} \tag{4.11}$$

If $f_i(S)$ equals zeros , it is ignored, to not introduce division errors. The lower the result the more similar the two distributions are.

4.4 Experiments

To examine the colour consistency of the descriptor methods, both intra and inter camera scenes for the descriptors were examined in detail. This is because the consistency of colour between consecutive frames is affected by different issues to that of object appearance consistencies inter camera.

4.4.1 Data Sequences

Two sets of image sequences were used for the evaluation of the methods. For the intra camera consistency, it was possible to make use of a popular groundtruthed data set from the CAVIAR project [1], the benchmarked set used is named "OneStopMoreEnter1". Examples from the dataset are shown in Figure 4.5.

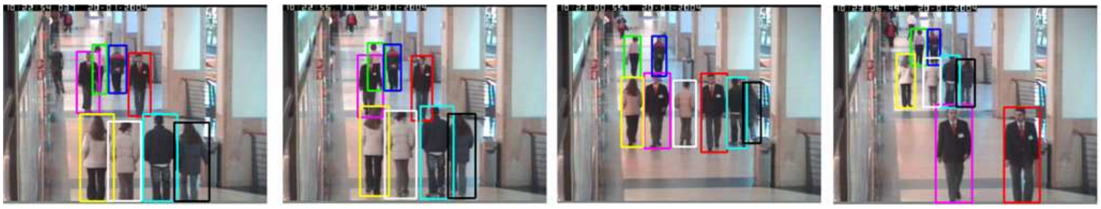


Figure 4.5: A selection of images from the "OneStopMoreEnter1" dataset from the CAVIAR [1] project.

The dataset is a low resolution fixed camera sequence with on average six people walking within the camera's field of view along a shopping centre, for 1123 frames at a frame rate of 25fps. The people interact and pass one another causing near total occlusions and pose changes while shadows and reflections are also present.

To examine inter camera object colour consistency there is no suitable publicly available groundtruthed dataset to use with multiple cameras. Therefore, a new sequence recorded using 4 non-overlapping cameras was used. The separate camera are multiplexed into a single time-synchronised video stream at a resolution of 320x240. Example frames of the inter camera sequence are shown in Figure 4.6.

The four cameras are cameras 1-4 in the set that are used within the real-time inter camera people tracker and are described in more detail in Chapter 6.

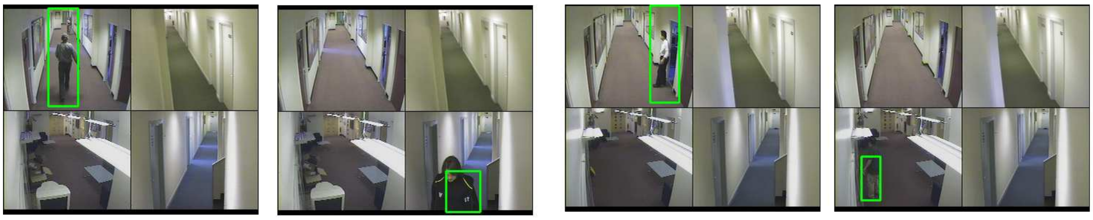


Figure 4.6: A selection of images of the inter camera sequence using 4-non overlapping cameras.

4.4.2 Object Appearance Consistency intra camera

To examine the consistency of colour between the frames t and $t+1$, a background mask is applied to the image and the foreground pixels within a groundtruthed rectangular kernel are used. These pixels intensities are used with the colour space and quantisation methods, to form an object descriptor, $f_R(t)$ of that person, R at time t. This is then repeated with all other m individuals visible in the frame. In the next frame, $t+1$ the same process is carried out. A true positive correlation is then computed between the descriptors for each person between frames t and $t+1$. This is the true positive similarity value for frame $t+1$ and is is summarised in equation 4.12.

$$\frac{1}{m}\sum_{i=0}^{m}\rho[f_i(t), f_i(t+1)] \tag{4.12}$$

In addition to computing the true positive correlation for a frame, the negative similarity or false positive rate is also computed. If it is excessively high, the descriptor will not be able to discriminate between true and false correlations effectively. The false positive rate is computed by finding the similarity between the query individual's object descriptor, $f_t(i)$ at frame t, and the m other candidates, at frame $t+1$.

$$\frac{1}{(m-1)m}\sum_{i=0}^{m}\sum_{j=0}^{m}\rho[f_i(t), f_i(t+1)] \quad where \quad i \neq j \tag{4.13}$$

These two tests are for all frames in the groundtruthed sequences. To analyse the results, the true positive and false positive values over the sequence were found.

The minimum, maximum, mean, and standard deviation were computed for each of the three techniques, colour space, similarity measure and quantisation size. These computations give a good indication of the compactness of the data and how close the true positive and false positive results are.

Another tool to measure the separability between the two classes of the true positive and false positive results is the T-test. The T-test can be used to determine whether the means are distinct, provided that the underlying distributions can be assumed to be normal. By using the T-test, which takes into account the spread of the data as well as the mean, class repeatability t can be found

$$t = \frac{\overline{x}_T - \overline{x}_C}{\sqrt{\frac{\sigma_T^2}{n_T} + \frac{\sigma_C^2}{n_C}}} \tag{4.14}$$

The higher the value of t, the greater the degree of class separability.

Parzen Windowing Intra camera

The initial experiments examined the effect of Parzen windows with increasing histogram bin size on the colour consistency of objects being tracked intra camera. The RGB colour space with histogram intersection for the measure of similarity were used. The appearance histogram had four different bin quantisation sizes; 3x3x3, 5x5x5 , 20x20x20, and 50x50x50. The Parzen window was fixed at 3x3x3. A large range of sizes was used to evaluate the ability of the techniques to cope with possible large variations in bin size.

Figure 4.7(a,c,e) shows the Histogram Intersection values through the video sequence for the true positive rate. It can be seen that In-Parzen windowing in Figure 4.7(b) performs the best with little difference between 3 and 5 bin sizes, while the distance to 20 and 50 bins is also lessened compared to no Parzen windowing (Figure 4.7(a)) and Post-Parzen windowing (Figure 4.7(e)). To examine this further a table of the minimum, maximum, mean, standard deviation, and

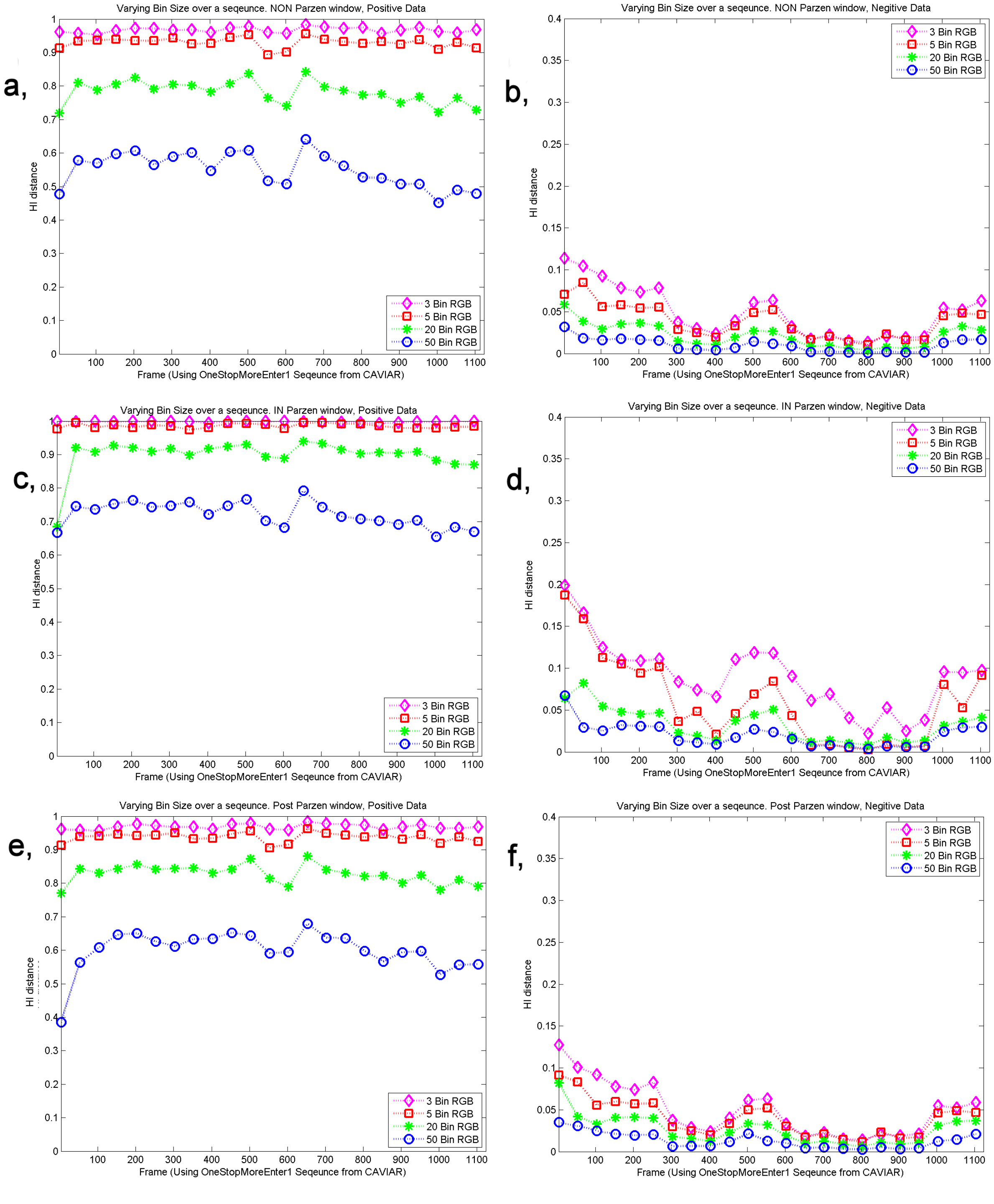


Figure 4.7: The Histogram Intersection for both true positive and false positive results for varying bin sizes using different methods on intra camera data. (a,c,e) show true positive results, for NON-Parzen windowing, In-Parzen windowing and Post-Parzen windowing respectively. (b,d,f) show the results of false positive examples using NON-Parzen windowing, In-Parzen windowing, Post-Parzen windowing respectively.

Table 4.1: Table of statical measures of the true positive results and negative results, from comparing the affect of bin size on, using a In-Parzen window, Post-Parzen window, and no Parzen window on intra camera data.

Stat	No-Parz Positive Results				No-Parz Negative Results			
Bin Size	3x3x3	5x5x5	20x20x20	50x50x50	3x3x3	5x5x5	20x20x20	50x50x50
Min	0.806	0.778	0.632	0.431	0.011	0.008	0.002	0.000
Max	1.000	0.999	0.998	0.997	0.147	0.118	0.089	0.063
Mean	0.968	0.935	0.795	0.583	0.048	0.036	0.020	0.010
S.D	0.018	0.025	0.065	0.127	0.030	0.021	0.015	0.010
T-test	141	141	91	108	n/a	n/a	n/a	n/a
	IN-Parz Positive Results				IN-Parz Negative Results			
Bin Size	3x3x3	5x5x5	20x20x20	50x50x50	3x3x3	5x5x5	20x20x20	50x50x50
Min	0.833	0.735	0.678	0.586	0.013	0.002	0.005	0.002
Max	1.000	1.000	0.995	0.998	0.199	0.196	0.115	0.088
Mean	0.996	0.986	0.914	0.740	0.088	0.057	0.030	0.019
S.D	0.012	0.017	0.031	0.081	0.040	0.049	0.020	0.014
T-test	133	121	131	78	n/a	n/a	n/a	n/a
	POST-Parz Positive Results				POST-Parz Negative Results			
Bin Size	3x3x3	5x5x5	20x20x20	50x50x50	3x3x3	5x5x5	20x20x20	50x50x50
Min	0.807	0.785	0.677	0.364	0.011	0.008	0.004	0.001
Max	1.000	0.999	0.998	0.971	0.148	0.127	0.098	0.069
Mean	0.970	0.941	0.837	0.632	0.048	0.098	0.025	0.013
S.D	0.018	0.023	0.053	0.101	0.029	0.022	0.016	0.010
T-test	143	133	104	62	n/a	n/a	n/a	n/a

the T-test of each method with each possible bin size is shown in table 4.4. As all the data are true positives, the Histogram Intersection should return a mean value of 1.000. However, as the bin quantisation size increases, the mean value over all the frames decreases. Thus, it makes it harder to differentiate between true and false positives. It can be seen that all techniques have a relatively high mean for both 3 and 5 bin quantisation levels. However, at a quantisation of 20x20x20, the mean histogram intersection without a Parzen window is only 0.795, a drop of 0.173 from the 3 bin level. With In-Parzen windowing, the mean is still high at 0.914, with a much smaller reduction of only 0.082. This lower reduction will mean that the bin size will have less effect on the performance of colour similarity intra camera if In-Parzen windowing is applied to the appearance histogram. Looking at the T-test, results for class separability between the true positive and false positive, it can be seen that the In-Parzen window maintains the highest values at the 20 and 50 bin sizes, therefore maintaining a distinction between the true and negative results.

Colour space Intra camera

The use of different colour spaces to represent the data on the sequence can have a large effect on the consistency of the object's colour intra camera. Three different colour spaces were investigated for use as the descriptor for object appearance intra camera. The HSV (Hue, Saturation, Value) model with quantisation (8x8x4), the colour Lookup table colour space (CLUT) [102][15] and traditional RGB quantisation (5x5x5) constructed with a In-Parzen windowing function applied.

Figure 4.8(a) shows the Histogram Intersection values through the video sequence for different colour models of the true positive rate. It can be seen that both the CLUT colour space and RGB quantisation have a similar performance over the sequence, with a lower performance for the HSV model intra camera. This

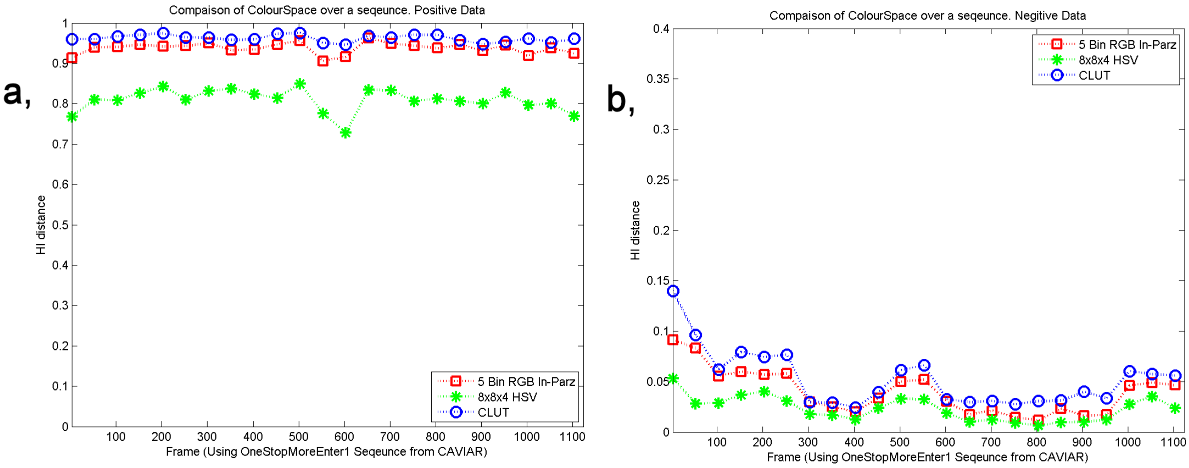


Figure 4.8: The Histogram Intersection for both true positive and false positive results for different colour models on intra camera data. (a) shows true positive results, for the three colour models (b) shows the false positive results for the colour models.

reduction in performance for HSV is partly due to the lack of a Parzen Window. Figure 4.8(b) shows the false positive rate over the sequence. The CLUT colour space has the highest false positive rate, with HSV the lowest, there is a trade off in performance due to less class separability. Figure 4.9 plots the true positive rate against the false positive rate with the ideal being the upper left corner. From this figure, it can be seen that all models have a degree of variance, with CLUT having the largest spread, while both the RGB and HSV results are more tightly centred. The T-test measure was used to compute the separability of the two classes, this is shown in table 4.2. It can be seen that the CLUT and RGB model have higher degree of separability partly due to the lower variance on the positive results from the ir higher T-test values.

In table 4.2, the mean and standard deviation is shown, and the CLUT colour space has the lowest standard deviation and highest mean for the true positive rate. However, as it has a lower T-test value, inter class separability is reduced. The lower computational cost similarity of results make the CLUT the best can-

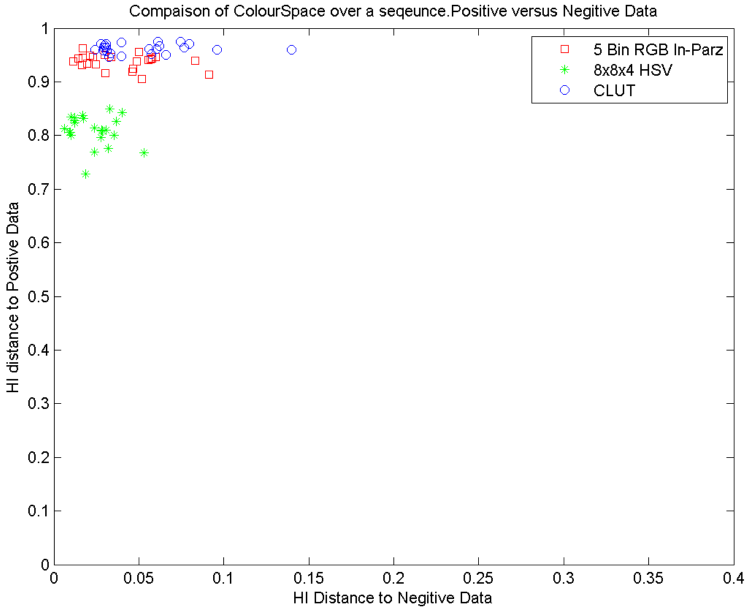


Figure 4.9: A Graph to show the true positive rate (y axis) against the false positive rate (x axis) for different colour models on intra camera data.

Table 4.2: Table of statical measures of the true positive results and negative results, comparing different colour spaces on intra camera data.

Stat	Positive Results			Negative Results		
Bin Size	HSV	CLUT	RGB	HSV	CLUT	RGB
Min	0.640	0.808	0.785	0.004	0.021	0.008
Max	0.999	0.999	0.999	0.093	0.140	0.127
Mean	0.821	0.964	0.941	0.022	0.051	0.038
S.D	0.057	0.019	0.023	0.014	0.024	0.022
T-test	100	148	143	n/a	n/a	n/a

didate to provide consistent object representation intra camera for a real time system.

Similarity Measures Intra camera

The measure of similarity between object appearance has a large effect on the performance of a system. Four different measures were examined, Histogram Intersection is a simple fast technique which provides a measure of similarity between histograms. The Bhattacharyya coefficient measures the degree of class separability. Mutual Information has basic spatial information encoded within the measure due to the joint probability, and Chi-square measures dissimilarly. The appearance histogram of the object was quantised using In-Parzen windowing into a 5x5x5 RGB colour space, with each similarity measure compared in Figure 4.10. It can be seen that the Bhattacharyya measure performs the most

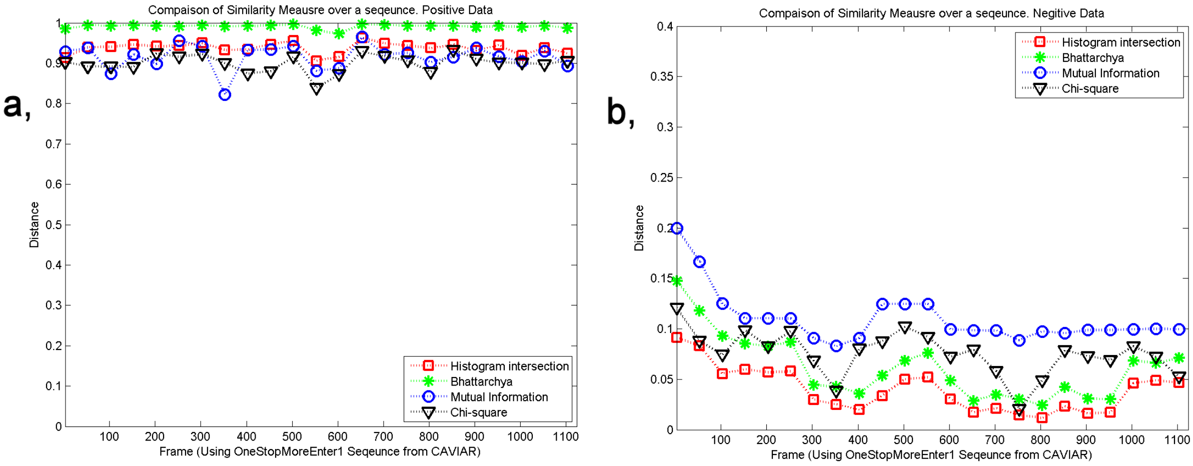


Figure 4.10: The Histogram Intersection for both true positive and negative results for different similarity measures on intra camera data. (a) shows true positive results, for the three colour models (b) shows negative results for the colour models.

consistently through the sequence, while the chi-square test and Mutual Infor-

Table 4.3: Table of statical measures of the true positive results and negative results, comparing different similarly measures.

Stat	Positive Results				Negative Results			
Bin Size	HI	Bhatt	MI	Chi	HI	Bhatt	MI	Chi
Min	0.785	0.828	0.691	0.743	0.008	0.013	0.051	0.012
Max	0.999	0.999	0.996	0.997	0.127	0.172	0.200	0.191
Mean	0.941	0.991	0.921	0.908	0.038	0.059	0.109	0.072
S.D	0.023	0.014	0.037	0.030	0.022	0.030	0.023	0.029
T-test	143	149	111	121	n/a	n/a	n/a	n/a

mation perform the worst. To investigate this further, the statical measures of the four techniques were examined in table 4.6. It can be seen that the Bhattacharyya coefficient measure has the lowest true positive standard deviation, and highest mean. However it also has a large standard deviation in the negative or false positive results. This is illustrated by Figure 4.11, where the Bhattacharyya results have a larger spread than Histogram Intersection. Despite this variance, the Bhattacharyya distance is the highest performing similarity measure, with the highest class separability and consistent true positive mean and therefore the best T-test score.

4.4.3 Object Appearance Consistency inter camera

To track the objects as they move inter camera, they are identified as an object of interest using the background segmentation mask. The person is then tracked with a dynamics model provided by a Kalman filter. They are continually tracked until they exit the camera's field of view. The appearance histogram of each individual is formed from the median appearance descriptor over the tracked

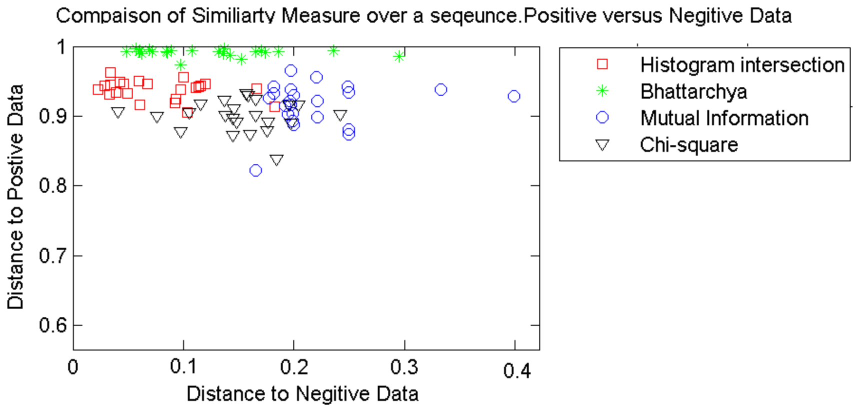


Figure 4.11: A Graph to show the true positive rate (y axis) against the false positive rate (x axis) for different similarity measures on intra camera data. (Note different axis scales)

frames. A median histogram is where each bin is found by taking the median value of that bin over the person's trajectory. After collecting all data it was manually groundtruthed to match tracks of the same person on the other cameras, and these became the true positive similarity matches. The negative, false positive matches were five other tracks taken at random from all the groundtruthed entries. The consistency of an objects appearance inter camera will depend on a number of different factors. The ability to cope with illumination and pose changes becomes far greater, and the need for good class separability illustrated through the T-test becomes important.

To examine the colour consistency inter camera, the dataset taken from the four non-overlapping camera setup was used. This contained 150 people tracked inter camera over a four hour period. The three different categories of techniques examined in the intra camera work were used again.

Parzen Windowing Inter Camera

The effect of increasing bin size on object colour consistency is examined. This is more challenging than tracking intra camera, as the illumination and the camera angle of the person will be dramatically different. Parzen windowing will partly reduce the effect of the illumination changes by expanding and spreading the quantised pixel values, to cover surrounding bins. This enables small illumination changes to be "coped with" efficiently. Four different bin sizes were examined 3x3x3, 5x5x5, 10x10x10, and 20x20x20, the appearance of objects inter camera was quantised into these bins using one of three techniques, In-Parzen windowing, Post-Parzen windowing and no-Parzen windowing. Figure 4.12 shows the results for varying bin size and quantisation techniques.

Looking at Figure 4.12(a,c,e) it can be seen that the overall performance is less compared to the intra camera work, with most values below 0.9 despite all comparisons being for the same person. Looking at the true positive value of Histogram Intersection for the people without a Parzen window, shown in Figure 4.12(a), it can be seen that some matches have a very low similarity measure of below 0.4. However, when In-Parzen windowing is applied during histogram construction as in Figure 4.12(c), two effects are seen. The first is that as the bin size of the histogram quantisation is increased, there is little performance degradation. This was expected and is similar to the results with the intra camera work in Figure 4.7(c). However, the overall performance of the matching is improved over Figure 4.12(a). These two improvements are due to the use of the kernel to smooth the data and thus ensure the quantised bins are not over or under filled. It also introduces variance to the data through the kernel function, thus allowing it to cope with the illumination variations in the appearance histograms. To examine this further, a table of the minimum, maximum, mean and standard deviation of each method with each possible bin size is shown in table 4.4.

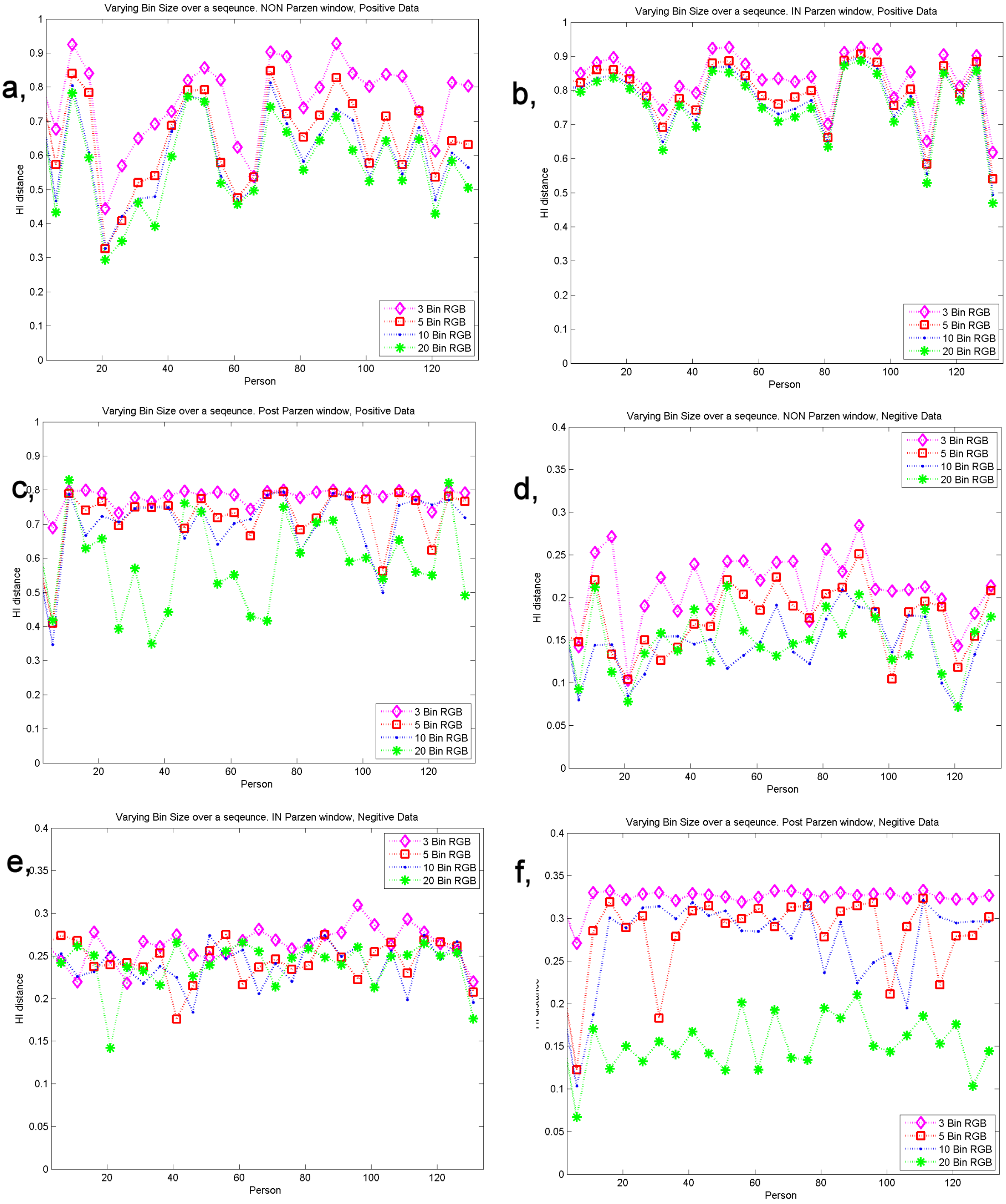


Figure 4.12: The Histogram Intersection for both true positive and negative results for varying bin sizes using different methods inter camera. (a,c,e) shows true positive results, for NON-Parzen windowing, In-Parzen windowing and Post-Parzen windowing repetitively. (b,d,f) shows the results of varying bin size of the negative examples using NON-Parzen windowing, In-Parzen windowing and Post-Parzen windowing respectively.

Table 4.4: Table of statical measures of the true positive results and negative results, from comparing the affect of bin size on, using a In-Parzen window, Post-Parzen window, and none for inter camera objects.

Stat	No-Parz Positive Results				No-Parz Negative Results			
Bin Size	3x3x3	5x5x5	10x10x10	20x20x20	3x3x3	5x5x5	10x10x10	20x20x20
Min	0.443	0.290	0.288	0.285	0.102	0.066	0.043	0.046
Max	0.985	0.965	0.941	0.928	0.284	0.253	0.220	0.218
Mean	0.785	0.771	0.619	0.593	0.219	0.178	0.154	0.153
S.D	0.105	0.132	0.136	0.139	0.036	0.035	0.033	0.034
T-test	51	49	38	35	n/a	n/a	n/a	n/a
	In-Parz Positive Results				In-Parz Negative Results			
Bin Size	3x3x3	5x5x5	10x10x10	20x20x20	3x3x3	5x5x5	10x10x10	20x20x20
Min	0.618	0.541	0.493	0.469	0.211	0.176	0.165	0.142
Max	0.942	0.925	0.910	0.901	0.309	0.296	0.288	0.280
Mean	0.840	0.802	0.780	0.765	0.267	0.249	0.240	0.235
S.D	0.076	0.088	0.095	0.100	0.020	0.022	0.025	0.025
T-test	62	56	53	50	n/a	n/a	n/a	n/a
	Post-Parz Positive Results				Post-Parz Negative Results			
Bin Size	3x3x3	5x5x5	10x10x10	20x20x20	3x3x3	5x5x5	10x10x10	20x20x20
Min	0.688	0.410	0.346	0.273	0.266	0.122	0.103	0.061
Max	0.800	0.799	0.798	0.955	0.333	0.326	0.325	0.241
Mean	0.786	0.737	0.718	0.605	0.324	0.285	0.275	0.157
S.D	0.017	0.069	0.078	0.148	0.011	0.040	0.043	0.038
T-test	93	46	43	35	n/a	n/a	n/a	n/a

Comparing In-Parzen windowing with both the Post and non- Parzen windowing methods across the quantisation bin sizes, it can be seen that the In-Parzen windowing provides a more consistent mean for increasing bin sizes in both true positive and false positive results. Using a Post-Parzen window on the appearance histogram improves the colour consistency compared to no window function, at the lower bin sizes of 3,5 and 10. However, when the 20x20x20 appearance histogram is used, it starts to fail in a similar nature to the Non-Parzen windowing method due to the fixed size Parzen window kernel.

Colour spaces Inter Camera

The choice of colour space can effect the performance of inter camera object correlation. This is excentuated as correlation assumes camera colour consistency. However, inter-camera, this is often corrupted due to the illumination differences of the cameras. Therefore, the method best suited to inter camera correlation will be able to compensate for varying camera colour responses. The same three colour spaces investigated perviously were used. The HSV (Hue, Saturation, Value) model with the quantisation (8x8x4), the CLUT colour space [102][15] and traditional RGB quantisation (5x5x5) with an In-Parzen window kernel.

Figure 4.13(a) shows the true positive rate of Histogram Intersection for different colour models. It can be seen that both the HSV and RGB quantisation have a similar performance over the tracked objects, with the CLUT colour space having a more erratic performance. The erratic performance is likely to be caused by the small dimensionality of the colour space causing some correlations to be underfitted and therefore matching well to many incorrect objects. However, Figure 4.13(b) shows the false positive rate with the CLUT colour space demonstrating the lowest false positive. Figure 4.14 plots the colour space's true positive rate against the false positive rate with the ideal being the upper left corner. Looking at this figure, it can be seen that the CLUT model has a larger degree

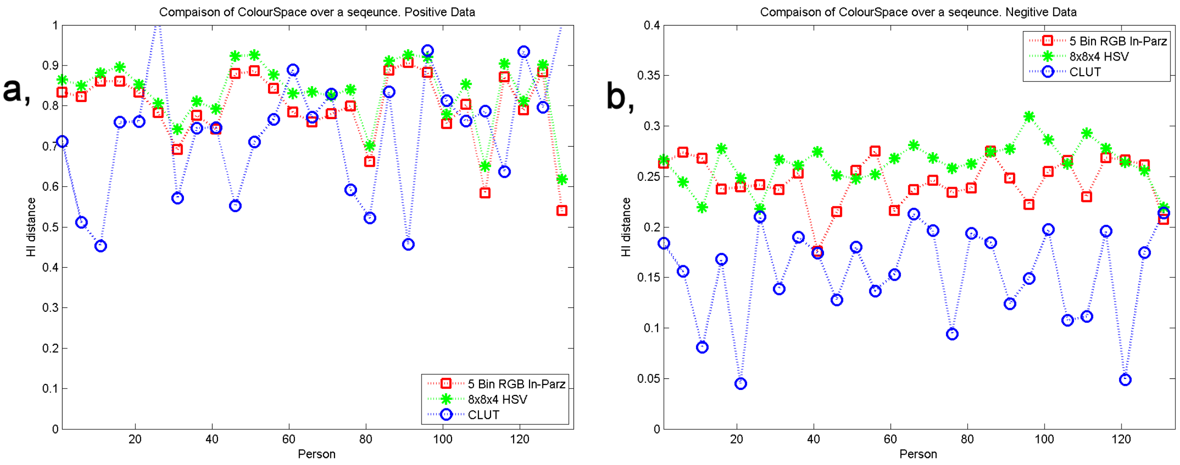


Figure 4.13: The Histogram Intersection for both true positive and negative results for different colour models for matching objects inter camera. (a) shows true positive results, for the three colour models (b) shows negative results for the colour models.

of variance however is closer to the ideal. While the HSV and RGB models have tighter distributions but a higher false positive rate.

To further examine these results, a table of statical summary is shown in Table 4.5. In this table, it is shown that despite the large standard deviation of the CLUT method, it has a comparable mean to that of the RGB and HSV models, while its much lower false positive mean, makes it an attractive colour space for inter camera tracking of objects.

Similarity Measures Inter Camera

As with the intra camera investigation the four similarity measures were examined, Histogram Intersection, the Bhattacharyya coefficient, chi-square and Mutual Information. The appearance histogram of the object was quantised using In-Parzen windowing into a 5x5x5 RGB colour space, and each similarity measure compared with both the true positive and false positive results. Figure 4.15 shows

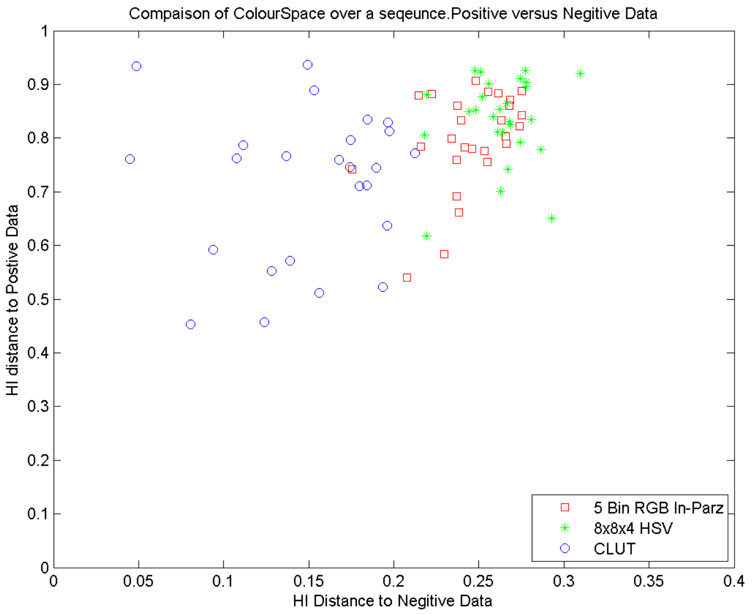


Figure 4.14: A Graph to show the true positive rate (y axis) against the false positive rate (x axis) for different colour models on inter camera data.

Table 4.5: Table of statical measures of the true positive results and negative results, comparing different colour spaces.

Stat	Positive Results			Negative Results		
Bin Size	HSV	CLUT	RGB	HSV	CLUT	RGB
Min	0.618	0.354	0.541	0.211	0.045	0.176
Max	0.942	0.999	0.925	0.309	0.264	0.296
Mean	0.839	0.792	0.801	0.267	0.166	0.249
S.D	0.077	0.166	0.088	0.020	0.050	0.022
T-test	52	45	55	n/a	n/a	n/a

the results for the groundtruthed objects that were tracked inter camera. It can

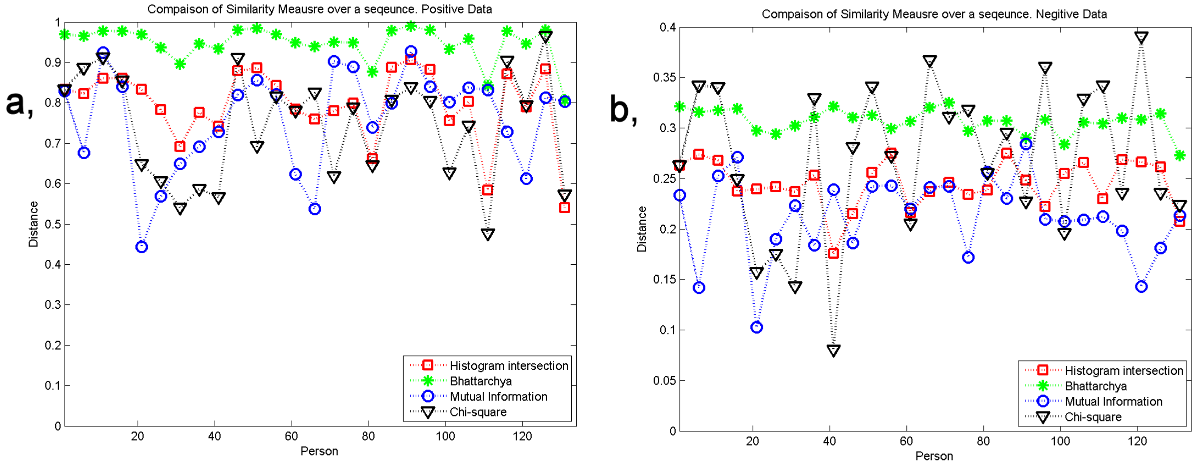


Figure 4.15: The Histogram Intersection for both true positive and negative results for different similarity measures for correlating objects inter camera. (a) shows true positive results, for the three similarity measures (b) shows negative results for the similarity measures.

be seen that with the true positive results in Figure 4.15(a), the Bhattacharyya has consistently high results, while Histogram Intersection,chi-square and Mutual Information failed for some of the tracked people. However, in Figure 4.15(b), the Bhattacharyya measure has a high false positive rate which could cause a degree of confusion between true and false positive identification of objects. In the statical summary of Table 4.6, the T-Test result for the Bhattacharyya measure shows that the degree of class separation between the true positive and false positive results is significatnly higher than the three other similarity measures.

The Bhattacharyya measure gives a higher degree of class separation between the true and false positive results and greater overall true positive performance on the inter camera tracking people. In addition, the technique is simple and low cost in terms of computation, making it idea for both on and offline processes.

Table 4.6: Table of statical measures of the true positive results and negative results, comparing different similarly measures inter camera.

Stat	Positive Results				Negative Results			
Bin Size	HI	Bhatt	MI	Chi	HI	Bhatt	MI	Chi
Min	0.541	0.807	0.443	0.440	0.176	0.255	0.102	0.081
Max	0.925	0.991	0.985	0.998	0.296	0.330	0.284	0.400
Mean	0.801	0.949	0.784	0.746	0.249	0.308	0.219	0.280
S.D	0.088	0.041	0.104	0.122	0.022	0.012	0.036	0.073
T-test	56	93	51	48	n/a	n/a	n/a	n/a

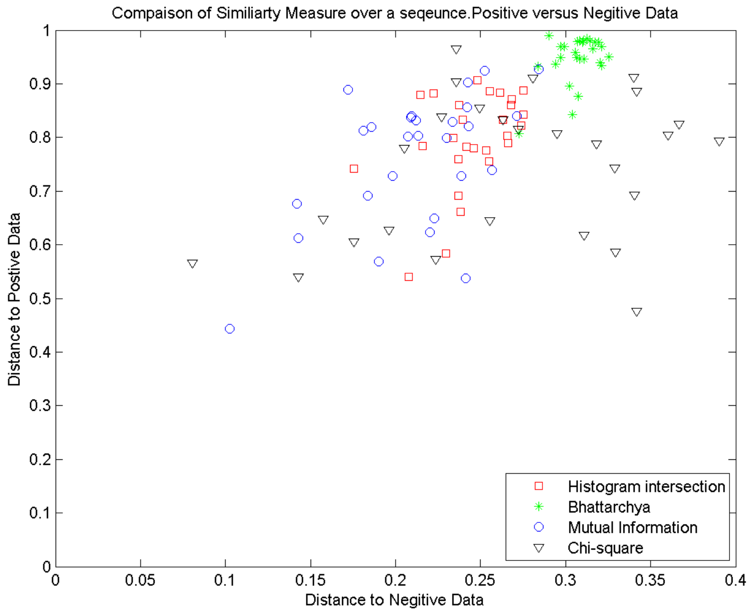


Figure 4.16: A Graph to show the true positive rate (y axis) against the false positive rate (x axis) for different colour models for inter camera data.

4.5 Conclusion

In total, six different investigations were performed to find the technique that gives the most consistent colour descriptor both intra and inter camera. Intra camera consistency is generally much easier as, between consecutive frames, there will be little variation in the appearance of the object and camera colour response is not an issue. However, at times of occlusion and during lighting changes due to shadows, the descriptor must compensate. The optimum histogram bin size is related to the number of samples. Therefore, an In-Parzen windowing function is applied to the data when forming the histogram. The reduction of performance as the bin size increases is greatly reduced for both intra and inter camera objects if this is done. The colour space in which the histogram is formed is important and from the results, the CLUT model had a high performance intra camera. For the inter camera all three colour spaces perform around the same level with similar class separation for their T-tests. However, it should be noted that both the HSV and CLUT models are not using Parzen windowing to achieve this, with far lower computation. Therefore the use of CLUT is advisable as it has the lowest cost with a similar performance to other colour spaces. The optimal measure of similarity to determine the correlation between histograms is the Bhattacharyya coefficient for both the intra camera in Figure 4.10(a) and inter camera in Figure 4.15(a). As it performs consistently better than Histogram Intersection, Chi-Square and Mutual Information at providing a consistent colour measure between frames, (intra camera), and across cameras, (inter camera).

This Section has examined the problem of providing a consistent appearance of objects moving both intra and inter camera. The use of the CLUT, with its arbitrary clustering of pixel values based on human perception, provides high performance with a low cost. The use of the Bhattacharyya coefficient measure is the most consistent at providing correlation between object's appearance histograms. However, if computational complexity is not a constraint, RGB quantisation with a Parzen window can provide a greater level of discrimative detail to maintain the identity of objects inter camera. These conclusions have been found through extensive testing on people tracked both intra and inter camera and shall inform the selection of techniques chosen in later sections.

Chapter 5

Tracking within crowded scenes

This chapter presents a solution to the problem of tracking people within crowded scenes. The aim is to maintain individual object identity through a crowd which contains complex interactions and heavy occlusions. In order to track multiple objects each object must be located and their identity labelled. An object's identity is simple to maintain when the tracked object is isolated. However, if interaction with other objects occurs, the identity can be lost or confused.

The approach uses the strengths of two separate methods; a global object detector and a localised frame-by-frame tracker. This is enhanced through the use of two priors learnt from the video scene during low activity periods, firstly, a temporal model of detections, to remove false positive detections. Secondly, the optical flow relationship of a moving person is learnt to provide increased accuracy of overlapping objects when tracking within occlusions.

A single camera with no explicit colour or environmental ground plane calibration is used. Results are compared to a standard tracking method and the groundtruth. Four different video sequences including two sequences from the CAVIAR dataset [1] are used to demonstrate the approach. They are all very challenging with low quality footage containing interactions, overlaps and occlusions between people. The results show that this technique performs better that a standard tracking method and can cope with challenging occlusions and group interactions.

5.1 Summary of the Algorithm

Multi-target tracking is a multiple stage process and is illustrated in Figure 5.1. Initially, a head and shoulders detector is applied to every frame to provide "seed" positions of visible individuals as shown in Figure 5.1a. The seed position is extended to cover the whole body using weak heuristics as shown in Figure 5.1b. Each seed (or body as in the image) is represented as an appearance model that is tracked locally over consecutive frames using a Mean Shift based appearance tracker to provide a short tracklet within the sequence (as shown in Figure 5.1c). When the appearance of the object of interest has changed significantly to the original, the tracklet is terminated. For a person walking through a scene, it is possible that they will have multiple head and shoulder detections for each frame. This will result is many short tracklets being formed, with each tracklet being split into its individual frame by frame tracks as shown in Figure 5.1d. A Viterbi style dynamic programming algorithm is used to find the optimum path through the frame-by-frame tracks based on a motion and appearance model. To reduce and remove unnecessary tracks, a model of the frame-by-frame spatial reappearance

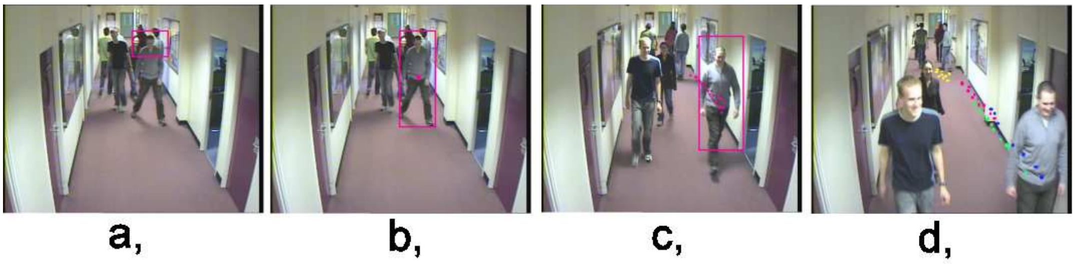


Figure 5.1: Simple Illustration of the stages in the tracking process

of head and shoulders detections is used to remove inconsistent detections. The lowest cost path through the sequence indicates the trajectory for each person.

5.2 Detecting Objects

In a crowded scene with overlapping people, traditional techniques such as a blob-based background segmentation give disappointing results. This is because blob based techniques require a clear separation between different object/ people. In addition, when people overlap or interact, much of the body outline is occluded. However, with the recording camera positioned above head height, as is often the case, the head and shoulders or upper torso are often visible even within a crowd. Therefore, a head and shoulder detector is used to produce seed locations of objects to be tracked. The detector is based on the one presented by Mikolajczyk *et al* [68]. The recognition technique uses a part based detector. Simple features are made up of local dominant orientations in the pixel's local neighbourhood. Figure 5.2(a) shows an example of the local features, with every three neighbouring orientations in a horizontal direction and a vertical direction being grouped as one. The locations are then quantised into a grid as shown in Figure 5.2(c) to form the features. During the learning process, the selected features from the object classes are represented in a single tree structure. The features are clustered with a method that produces a hierarchical tree of clusters. Figure 5.3(a) illustrates the tree representation, while Figure 5.3(b) shows the use of a feature codebook. The codebook is a list of all possible features, which are then accessible to multiple tree nodes. This allows both the motorcycle and push bike to use the same circular features, in conjunction with other unique features. The use of a hierarchical clustering allows for increased computation efficiency, while being robust to minor occlusions of the individual features of the object. A detector trained to detect faces would be unsuitable as faces are

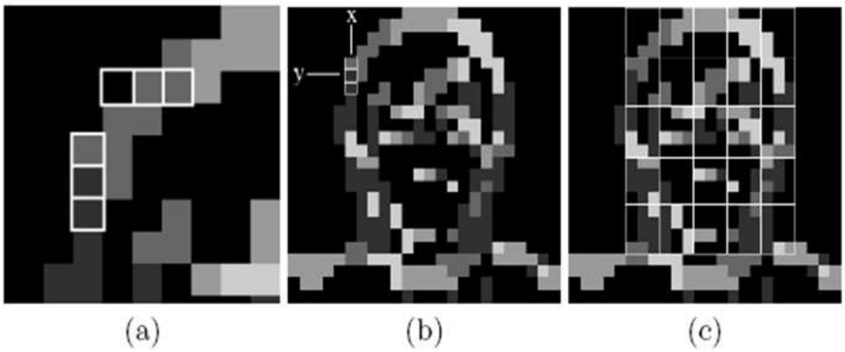


Figure 5.2: Local features; (a) Two groups of local orientations, (b) Location of features on object, (c) A grid of quantised locations

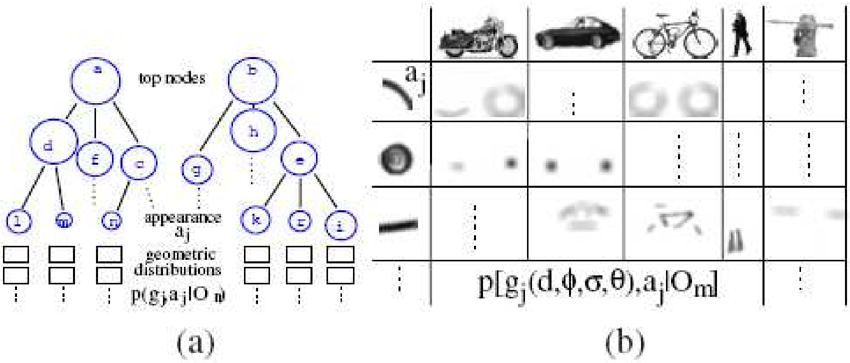


Figure 5.3: (a) Hierarchical structure. (b) Codebook representation. Appearance clusters (left column) and their geometric distributions for different object classes. [68]

often not looking directly at the camera, and thus would cause a false negative. Likewise, a full body detector would be unsuitable as parts of the body are often heavily occluded. An example frame from a sequence where a face or full body detector would fail due to people facing away from the camera and occluded is shown in Figure 5.4

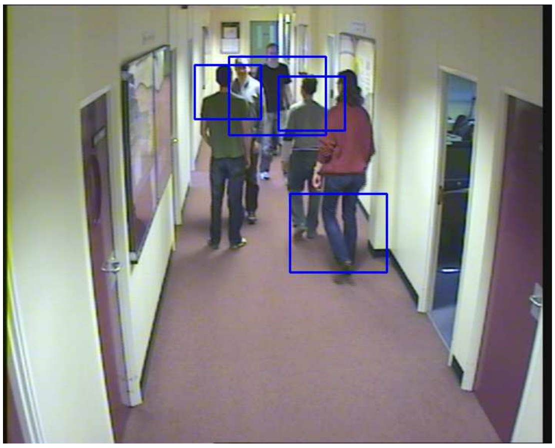


Figure 5.4: This shows a frame in a sequence showing the main head and shoulder detections. Note there are both true positive and false positive detections.

In order to train the detector, 1200 head and shoulder examples were manually segmented from images from the internet. The examples contained centred people facing in all directions with respect to the camera as shown in Figure 5.5. In classification, each detection has a confidence value. If the confidence value is greater

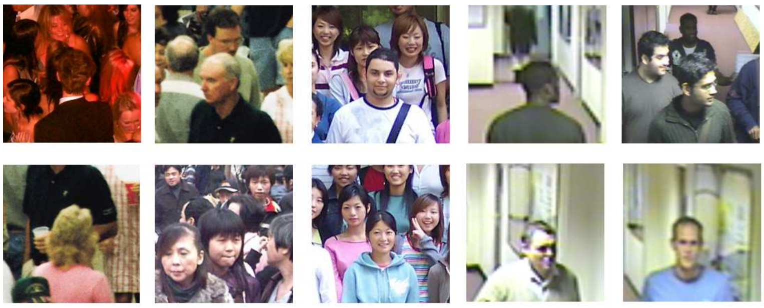


Figure 5.5: Positive training examples for the head and shoulder detector

than a threshold, the detection is used. To define this threshold, two unseen sequences of groups of people had their body outline manually groundtruthed. The detector was applied to the two sequences, and the confidence value threshold of a detection was varied, producing a ROC curve. Figure 5.6 shows the response by the detector to the video sequences, the threshold to give 400 false positives is used in the detector. The chosen threshold gives a higher false positive rate than is usual within the object recognition and classification field. This allows for a

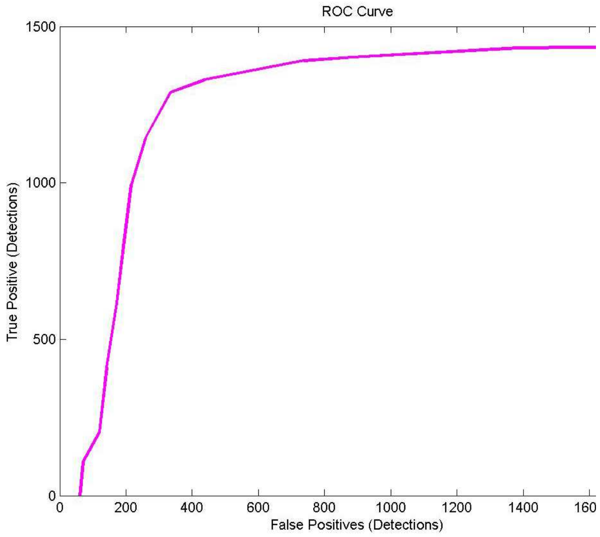


Figure 5.6: This shows an ROC curve for the head and shoulder detector used with the sequence of a crowded scene

greater number of detections for the *harder* objects such as people walking away from the camera or those who have a large proportion of their outline occluded. These false positive detections will be identified as outliers to the motion and appearance models, and later discarded. Each head and shoulder detection has a rectangular kernel that is centred on the detection. The height of the kernel is then used as a weak heuristic to estimate the total size of the person to cover their whole body.

5.3 Learning Models of Motion

When people move within a cameras field of view they often take similar routes to reach a door or avoid fixed obstacles, this means people will move in similar motion patterns on individual cameras over time. This key observation can be exploited to learn during periods of low activity when tracking is simpler and use this during busy periods to help disambiguate. There are two priors that are learnt during these low activity periods.

5.3.1 Learning Head and Shoulder Detection Relationships

The head and shoulder detector produces a large number of false positive detections, these produce Mean Shift tracklets that can corrupt tracking. Therefore a method to identify the false positive detections is used, allowing false tracklets to be ignored. A single model of detection over time and space is learnt for each camera viewpoint. This can be used to predict future head and shoulder positions and therefore ignore false positive detections based upon their likelihood of fitting the model.

A one hour sequence of video was used to learn the relationship. For each sequence, each detection is compared to all previous detections with respect to the time elapsed and the x, y pixel difference to the original detection. A pdf in dx, dy, dy is then incremented with a In-Parzen window blur for each possible pair of detections. For a set of W head and shoulder detections $D \epsilon \{D(1), ...D(W)\}$, equation 5.1 computes the frequency of bins in the histogram representing the reappearance of head and shoulder detections.

$$f_{xy} = \sum_{S=1}^{W} \sum_{R=1}^{W} \eta((D_x(S) - D_x(R), D_y(S) - D_y(R), t(S) - t(R)), I\sigma^2)$$
$$where \quad 0 \leq (t(S) - t(R)) < T \tag{5.1}$$

Where $D(R)$ is the current head and shoulder detection and $D(S)$ is a previous detection. $D_x(S) - D_x(R)$ and $D_y(S) - D_y(R)$ are the difference in position in both x and y respectively between detections S and R. t is the time of the detection, with T set as a maximum reappearance period to limit the temporal length of the prior, this is commonly 100 frames. $\eta(\overline{V}\epsilon\Re^3, I\sigma^2)$ represents a 3D gaussian kernel positioned at V with spherical co-variance σ^2. An elliptical co-variance in time could also have been used, this would allow independence between time and space, however it was found to not be required. This process is then repeated for all other detections in the sequence, forming a model of the head and shoulder detections over time. Given a head and shoulder detection $D(R)$ its prior reappearance probability over time can be modelled by equation 5.2

$$p(D_t(S)|D(R)) = \frac{f_t^{R|S}}{W^2} \tag{5.2}$$

Figure 5.7 shows the reappearance probability of a detection $p(D_t(S)|D(R))$ where (a) $t = 5$ frames and (b) $t = 100$ frames. The centre of the likelihood image is where the original detection occurred. Notice how this encodes domain knowledge for the camera, as the slight diagonal trend corresponds to the off centre placement of the camera and therefore the trend for people to move diagonally. By 100 frames there is a more dispersed distribution, this is due to the increased uncertainty of predicting further into the future.

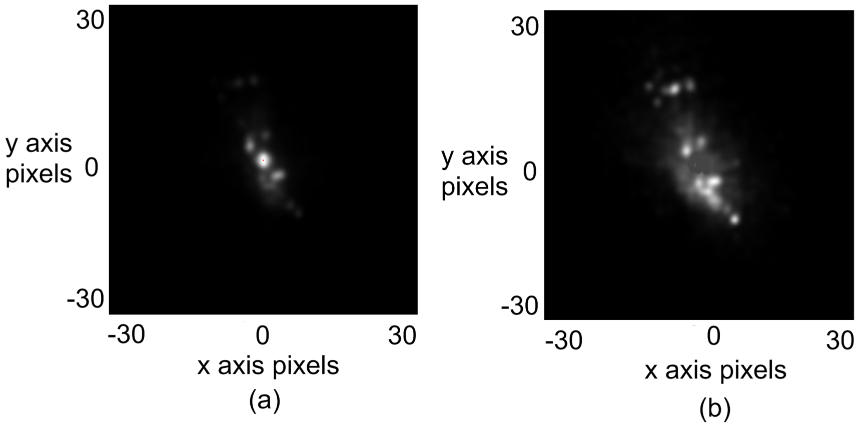


Figure 5.7: Figure (a) shows the conditional probability of a head and shoulder detection given a detection occurred 5 frames ago in the centre. Figure (b) shows the conditional probability of a new head and shoulder detection given a detection occurred 100 frames ago in the centre.

5.3.2 Optical Flow Accuracy Prior

Another problem occurs when objects occlude each other. While overall tracking is robust to this, the occlusion can cause the Mean Shift tracker to be centred in-correctly on the two objects. This means the overall accuracy of the tracker is reduced. To improve this, the optical flow of the pixels is used. Optical flow can be applied to the consecutive frames of a person moving through the image. Optical flow is an approximation of the local image motion based upon local derivatives in a given sequence of images. It specifies how much each image pixel moves between adjacent images. The basic assumption for the optical flow calculation is that of the conservation of pixel intensity. It is assumed that the intensity, or colour, of the objects has not changed significantly between the two frames. The Lucas-Kanade method [63] is used for computing the pixel based optical flow. Figure 5.8 shows the optical flow field of the foreground area of the head and shoulders of a person from the CAVIAR dataset walking away from the camera. Despite the poor quality compressed image making the optical

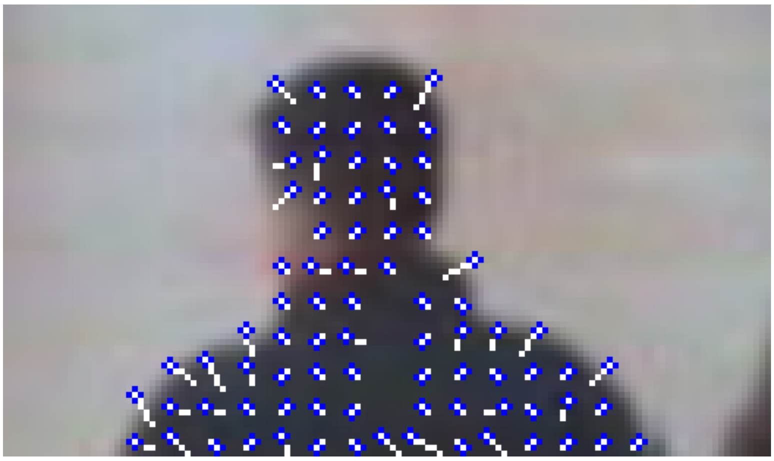


Figure 5.8: The Optical Flow field of a tracked person

flow field noisy, the largest vectors are generally around the edge of the person, while within the person the optical flow vectors have a smaller magnitude. In addition, the direction of the vectors around the edges is in the same direction as the tracked object's global motion (in Figure 5.8 this is upwards). If two or more people overlap, the optical flow can be used to identify the ownership of the pixels. In addition to being used to resolve overlapping objects, it can also be used to optimise the bounding box of tracked people, when the bounding box of person is poorly centred on the actual object of interest. This is often due to the head and shoulder detector location, the scale not being optimal, or the Mean Shift tracker drifting.

Learning the pixel motion model

A model of the pixel motion within an objects bounding box is learnt to aid pixel labelling when objects overlap. Each pixel within the fixed size bounding box has a 2D histogram. The 2D histogram is the velocity probability of optical flow for each pixel. During the training phase, when the head and shoulder detector relationship prior described in the previous section is learnt, optical flow is applied to each frame. To remove noise from the optical flow image, the background segmentation mask is used to segment only the foreground pixels. The optical flow of an objects foreground pixels within this bounding box region is found with respect to the previous frame and the bounding box of the object is normalised to a predefined size.

For each pixel within the normalised bounding box, the pixel's optical flow value is quantised and added to the 2D histogram of the optical flow at that pixel point. An In-Parzen window blur is applied to reduce quantisation effects. This is repeated with all foreground pixels in the object's bounding box.

The process is repeated for all other head and shoulder bounding boxes. Over the sequence, a generic prior will be learnt from the optical flow. Figure 5.9 shows images of the resulting likelihood model of 2D histograms representing the Optical Flow. Figure 5.9(a) shows the overall bounding box with each pixel having a separate 2D probability of optical flow for that pixel. It can be seen that the foreground region of a person is learnt. Figure 5.9(b) and (c) show enlarged parts of (a), with each separate 2D histogram enlarged. Figure 5.9(d) shows a single 2D histogram, that corresponds to a single pixel within the overall bounding box, together with the axis limits.

5.4 Forming Object Tracks

The tracking system takes all the head and shoulder detection locations, with the aim of extending their trajectory further through the sequence. The area within the object's kernel is used to form an m bin histogram appearance model, q_u where $u = 1...m$. The object's movement over the sequence is tracked over time using a Mean Shift tracker with Kalman Filter prediction to initialise the Mean Shift tracker on a frame-by-frame basis.

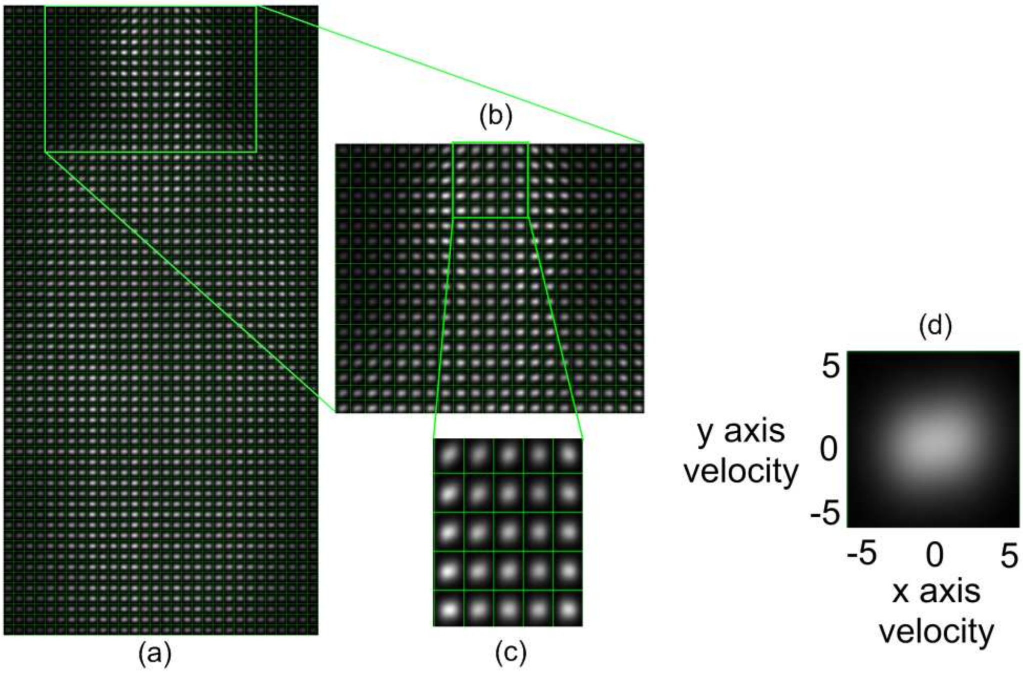


Figure 5.9: The learnt likelihood model of the optical flow of Optical Flow field, with a zoomed area around the head region

5.4.1 Kalman Filter Prediction with Mean Shift

A Kalman Filter is used in conjunction with the Mean Shift utilising the assumption that over time people walk with a constant velocity. This allows dynamics to be introduced to the Mean Shift which otherwise only optimises tracking based upon the local moment of a colour distribution. A constant-velocity model with a white noise drift is assumed for the Kalman Filter. For details, Section 3.3 in Chapter 3 provides more detail about the update equations. For a new frame, the Kalman Filter will predict the centre of each objects's kernel. This area is then used by the Mean Shift as the initial search region for the object. The Mean Shift will then optimise this prediction through iterations of the Mean Shift procedure until convergence. The object's Kalman Filter is then corrected with the converged Mean Shift measurement. This correction is then used by the object's Kalman Filter to predict the position of the object's kernel on the following frame which is used again as the Mean Shift search region. For an in depth discussion of Mean Shift see Section 3.2 in Chapter 3

5.4.2 Track Termination Criteria

Although the Mean Shift with velocity prediction will cope with minor occlusions or lighting changes, large scale appearance changes to the kernel of the tracked object can cause Mean Shift to fail. Therefore, a termination criteria is applied to determine when the Mean Shift kernel has failed and is no longer tracking the original object. If multiple short but significant tracks of the same object are found, these can then be combined together to produce the full trajectory. The termination criteria is based on a likelihood ratio based on training data. The likelihood ratio is based on two learnt cumulative frequency histograms. The two histograms are formed of the probability of positive and negative objects respectively. To form the cumulative probability, the groundtruthed training sequence used for the Models of Motion in Section 5.3 are employed. For each groundtruthed object track, the similarity of each separate individual inter frame is computed by the Bhattacharyya coefficient and is used to populate the positive cumulative frequency histogram. While the negative probability is formed from the the similarity of an object with other objects on the following frame. Figure 5.10 shows the positive and negative cumulative frequency histograms. This cumulative histogram can then be used as a probability of the correlation between two objects being a positive or negative correlation.

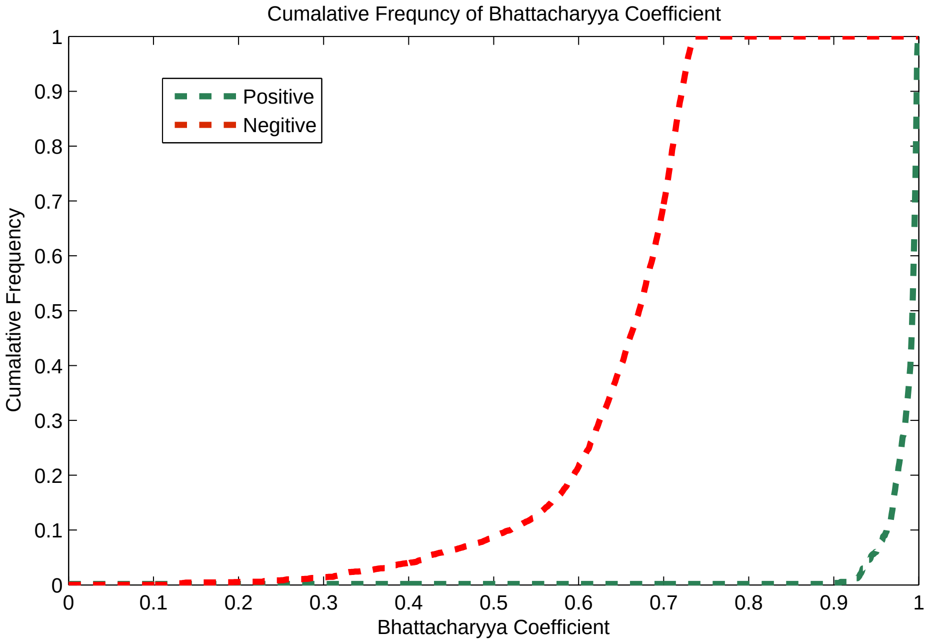


Figure 5.10: The Cumulative frequency probability for object similarity for true and negative objects.

Within the Mean Shift tracker, the reference appearance model $\psi*$ is updated every 25 frames, and subsequent frames are then correlated to this appearance model. This correlation uses the learnt cumulative frequency histograms to provide a ratio of similarity. This is the likelihood ratio of the tracked kernel area, ψ_t at frame t, being *more* similar to the appearance model of the track, $\psi*$, than *not* similar to the appearance model. This is shown in equation 5.3

$$L(K_t|\psi*) = \frac{P(\psi_t|\psi*)}{P(\psi_t|\overline{\psi*})} \tag{5.3}$$

where $P(\psi_t|\psi*)$ is the probability of the kernel ψ_t being the appearance model for the track and $P(\psi_t|\overline{\psi*})$ is the probability of the kernel not being the appearance model for the track. These are calculated as the cumulative probability, of the Bhattacharyya similarity coefficient shown in equation 5.4, where $i = \{0.01, 0.02, 0.03, ...\rho[\psi_t, \psi*]\}$. Where positive examples are used for $P(\psi_t|\psi*)$ and negative groundtruthed examples used for $P(\psi_t|\overline{\psi*})$.

$$P(\psi_t|\psi*) = \sum_{i=0}^{\rho[\psi_t,\psi*]} F_i(\psi) \tag{5.4}$$

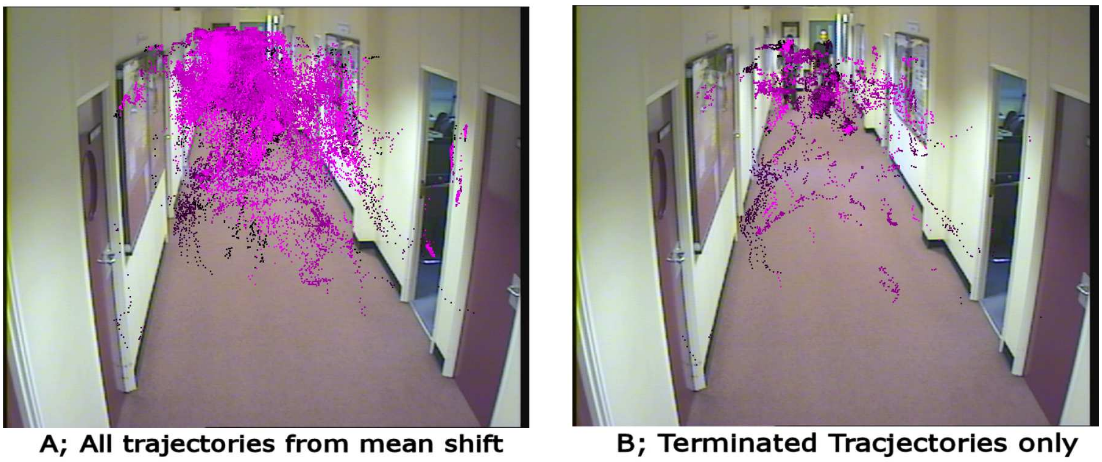
where $\rho[\psi_t, \psi*]$ is the Bhattacharyya similarity coefficient between ψ_t and $\psi*$, (shown in equation 5.5) and $f_i(\psi)$ is the normalised frequency histogram of the Bhattacharyya coefficient over a number of ground-truthed object sequences.

$$\rho[\psi_t, \psi*] = \sum_{u=1}^{m} \sqrt{\psi_u * \psi_u*} \tag{5.5}$$

If the likelihood ratio $L(K_t|\psi*)$ is less than 1 the trajectory track is terminated, forming a short tracklet, $\{ST_{t\,start}, ..., ST_{t\,end}\}$. Where ST_t is the state at time t represented by the colour appearance model (colour histogram) $ST_{\psi t}$ and motion model ST_{xy} (Kalman filter state). Figure 5.11 shows how the tracks are significantly reduced once the termination criteria is applied.

5.4.3 Kernel Scale Adaption

Within the Mean Shift tracker proposed by Comaniciu [24], there is a simple scale adaption technique that adjusts the kernel size based on maximising the Bhattacharyya coefficient for different kernel sizes. This adapts the scale of the kernel well for sequences with little or no occlusions, although as the occlusion



A; All trajectories from mean shift

B; Terminated Tracjectories only

Figure 5.11: A, shows all the trajectories from the seeded Mean Shift tracker of a crowded video sequence, B shows the trajectories, where the tracks have been terminated using the termination criteria in Section 5.11.

level increases, the overall tracking performance drops. This is because the optimised scale doesn't take into account that part of the person could be occluded, and will therefore only optimise on the visible part of the object. This can cause misleading and incorrect scale adaption. Within this work there is no need for an adaptive scale within the Mean Shift tracker as the head and shoulder detector determines the scale of the kernel. This means that as the person increases in size the scale in the head and shoulder detection will increase ensuring the trajectories bounding box will adapt in size as required. This means that the overall trajectory of an object has a piece-wise linear approximation to scale changes when the short tracklets are recombined in Section 5.5.

5.5 Combining Tracklets into Trajectories

The Mean Shift kernel tracker produces short tracklets for each person visible within the video sequence. Each person will have multiple tracklets over time, and the aim of this section is to find the optimum path through the complete sequence. This is solved through a dynamic programming algorithm. Dynamic programming is designed to find the shortest path in a graph using a optimal substructure. It works by breaking the problem of finding the lowest cost path through the node states into smaller sub problems. The are many algorithms used to find the shortest path including Dijstra's algorithm [27] [113], The Viterbi algorithm [108], and The Generalized Bellman-Ford [9]. The Dijstra's algorithm is used for best-first ordering, in that it uses an evaluation function and always chooses the next step to be that with the best score. While the Viterbi uses a topological approach, and The Generalized Bellman-Ford is a combination of the two, together with the ability to use negative weighting on paths. For this work the Viterbi algorithm was chosen.

The Viterbi algorithm operates on a state machine assumption. That is, at any time the system we are modelling is in some state. There are a finite number of states, however large, that can be listed. Each state is represented as a node. Multiple sequences of states (paths) can lead to a given state, but one is the most likely path to that state, called the "survivor path". This is a fundamental assumption of the algorithm, because the algorithm will examine all possible paths leading to a state and only keep the one most likely. This way the algorithm does not have to keep track of all possible paths, only one per state.

Each tracklet $ST_\tau = \{ST_{t\,start}, ..., ST_{t\,end}\}$ produced by the Mean Shift tracker forms a partial row in the state matrix ST between the time indexes $_{t\,start}$ and $_{t\,end}$, where τ is a state for a frame. Each state τ is a node within the overall *graph* or state matrix. The objective is to find the optimal path through this state matrix that maximises the likelihood of the trajectory for the reference object REF which has a state of ST_{REF}. This is found using three cost functions that are used to weigh the paths between nodes. There are three likelihoods; L_{Ref}, L_{App} and L_{KF} and the learnt detection reappearance model from Section 5.3.1

L_{loc}. These four likelihoods are expanded below.

The appearance similarity L_{Ref} between the State appearance $ST_{\psi t}$and the reference appearance model $ST_{\psi REF}$ is found using the Bhattacharyya similarity coefficient 5.5.

$$L_{Ref} = \rho[ST_{\psi,t}, ST_{\psi,REF}] \tag{5.6}$$

This provides a constraint ensuring the trajectory will stay visually similar to the original person's reference image. Between frames t and $t+1$ the appearance similarity L_{app} between the states is computed using the Bhattacharyya similarity coefficient 5.5.

$$L_{app} = \rho[ST_{\psi,t}, ST_{\psi,t-1}] \tag{5.7}$$

A motion model is computed for each state trajectory adding a non appearance based constraint.

$$L_{KF} = p(S_{xy,t}|S_{xy,t-1}) \tag{5.8}$$

The Kalman Filter [109] in the Mean Shift kernel tracking in Section 5.4 computes predicted positions $\{ST_{xy,t}, ST_{xy,t+T}\}$ allowing each tracklet to be artificially extended. T is typically set to 4 seconds, i.e. 100 frames. The extended track allows for a short overlap between separate tracks of the same object, reducing sharps jumps between tracks and overcoming occlusion. The Mahalanobis similarity measure is used to find the difference in position between the predicted state positions and current state positions. The detection reappearance model $p(D_{tj}|D_{ti})$ from equation 5.2 is used as a prior to constrain the other similarity measures. The likelihood of the state at frame t is computed from the path in frames t to $t-T$.

$$L_{loc} = \prod_{i=1}^{T} p(D_t|D_{t-i}) \tag{5.9}$$

Dynamic programming is then used to select the optimal path that maximises the objective function,

$$\Phi(l) = \max_{\tau}\{L_{Ref}L_{app}L_{KF}L_{loc}\}\Phi_{t-1}(\tau) \tag{5.10}$$

where there are τ possible states for a frame. This can be visualised as a trellis diagram, as shown in Figure 5.12. There are 4 state tracks shown in Figure 5.12,

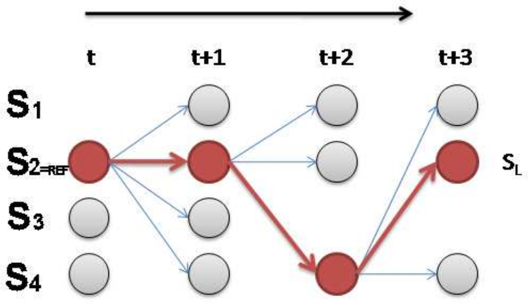


Figure 5.12: A visualisation of the Trellis diagram for find the lowest cost path through the video sequence.

with the S_{REF} selected as the starting image of the person to track. Then at each time interval, t, all possible paths from the current state are examined and the cost is maximised to find the next most likely state.

5.5.1 Multiple Paths

The recursive algorithm in equation 5.10 maximises a single best path. Therefore should two or more trajectory states maximise to the same destination state, the state transition with the highest likelihood will use that destination state. The remaining trajectories will re-evaluate the remaining states, and repeat the process of the maximising the state transitions. This allows multiple trajectories within the state transition trellis design while achieving multiple near optimal paths through it.

5.5.2 Improving Accuracy

The trajectory computed through the sequence for an individual will not be completely centred on the individual at times, with minor "jumps" when the best computed path changes to other tracklets. To remove these the *a priori* learnt optical flow likelihood can be used. The prior describes the inter frame motion of pixels in an objects bounding box. It can localise and improve the accuracy of the trajectory on a frame-by-frame basis by reducing jumps caused by occlusions or Mean Shift tracker failure. Each pixel in the sequence has an optical flow value, this is used to compute a likelihood of that pixel belonging to a specific object. When objects overlap, a pixel will have multiple likelihoods for all the overlapping objects. The pixel is then assigned the identity of object with the highest likelihood. A Kalman filter motion model is also applied to the centroid of the tracking kernel to smooth out any jitters, assuming a constant motion model.

5.6 Experiments

To examine the effectiveness of the multiple person tracker, four separate sequences containing up to 8 people walking in crowded scenes were examined. This section shows the tracking results on people walking and interacting in indoor corridors.

5.6.1 Data Sequences

Following the experiments of Chapter 4, concerning colour consistency of object intra camera. Figure 4.8 in Section 4.4.2 shows that both CLUT and quantised RGB with an In-Parzen window have a high consistency of colour intra camera. Both would be well suited to cope with the lighting changes that occur in non uniformly lit areas. However, quantised RGB is used here as there are no real-time constraints and the descriptor can hold more discrimative information about the object than the 11 bin CLUT method. Two different sources were used to provide a total of four sequences to test the method. Two were sourced from a newly created groundtruthed dataset and two from the CAVIAR video sequences.

Dataset 1

A dataset was recorded, (Dataset 1) as it was found that the publicly available benchmarked data sets, including the CAVIAR [1] and PETS [3] video sequences lacked heavy occlusions and therefore were not challenging enough. Two new sequences were groundtruthed and used. The first sequence consists of seven people walking towards and away from the camera. There is overlap between the individuals and complete occlusions occur within the sequence. The second video is similar but many of the individuals in the sequence stop mid way through the sequence and there are a greater number of interactions and heavy occlusions. This is designed to show the limitations of motion based trackers such as the Kalman Filter, and appearance based trackers such as Mean Shift. Both sequences are very challenging due to a number of reasons. The lighting used is non uniform, this affects the mean shift tracker and head and shoulder detector. There is a large field of view causing a large scale variation as people move along the corridor. The camera is mounted less than a metre above head height causing many occlusions. The sequences were captured using a single surveillance style PAL resolution camera at 25fps. It is a conventional low cost camera, with poor colour response and high pixel noise levels.

CAVIAR

To provide a degree of comparison to other techniques, two commonly used sequences from the CAVIAR [1] project were used. Only two sequences are used as the majority of sequences contain few occlusions or interactions making them too simple for most state of the art tracking techniques. The two sequences were taken separately from the sequence labelled “OneStopMoreEnter1”. The sequences are low quality with a very reflective white floor and a wide field of view. Both sequences have people overlapping and walking away from each other, some full occlusion also occurs.

The sequences were ground truthed and the approach compared with the Mean Shift algorithm [24] for each of the four sequences. The Results are presented both qualitatively with bounding boxes indicating the people tracked and quantitatively through graphs and tables. To assess performance, the Euclidean distance between the centre of the groundtruth's bounding box and that of the computed trajectory is calculated to give a distance error. The percentage of overlap between the groundtruth and computed trajectory bounding box gives an indication of correct scaling and accuracy. The newly created dataset sequences are shown first, and then the two CAVIAR sequences. Then the improvements using the optical flow prior and a Kalman filter are presented separately on the CAVIAR sequences.

5.6.2 Dataset 1

People in Video 1 were tracked using three possible methods. The state of the art Mean Shift algorithm, the main method in this chapter called a tracklet tracker introduced earlier, and the tracklet tracker with the additional head and shoulder detection prior outlined in Section 5.3.1 providing a predictive motion model. Figure 5.13 shows the mean euclidean distance error and mean overlap per frame with the groundtruth kernel. Figure 5.13(a) shows that all three methods have similar average Euclidean difference per frame, however Figure 5.13(b) shows that the tracklet tracker with the head and shoulder prior is much more accurate at tracking the kernel sizes, this is shown by the stable overlap percentage through the sequence. While Mean Shift overlap percentages drops after heavy occlusion occurs midway through the sequence.

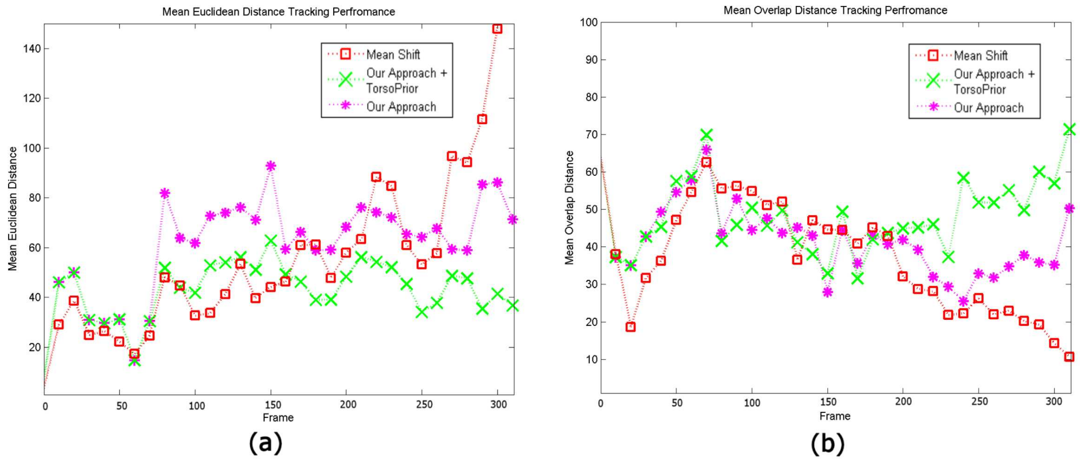


Figure 5.13: **Video 1:** Figure (a) shows the mean Euclidean distance difference per frame for the tracklet tracker with the learnt prior and without and also Mean Shift (less is better). Figure (b) shows the mean overlap per frame for the tracklet tracker with and without the prior and Mean Shift (more is better).

To provide more detail Figure 5.14 shows the results of every 10 frames for people tracked within the sequence for both the tracklet tracker with head and shoulder prior and the Mean Shift.

Figure 5.14(a) and Figure 5.14(b) shows the Euclidean distance error between the bounding boxes' centre computed trajectory path and that of the groundtruths' bounding box centre over video 1. It can be seen that overall the Tracklet Tracker with Prior in Figure 5.14(a) minimises the Euclidean error distance compared to the Mean Shift tracker Figure 5.14(c). While both trackers work well at the beginning of the sequence when there is little occlusion or interaction, towards the end of the sequence the Mean Shift tracker fails on a number of people as the interactions become more complex. Occlusions are solved by finding alternative paths through the sequence. Figure 5.14(b) shows the overlap of the Tracklet Tracker with prior computed trajectories bounding box with that of the groundtruth, with Figure 5.14(d) repeating this for a Mean Shift tracker. Figure 5.15 gives a

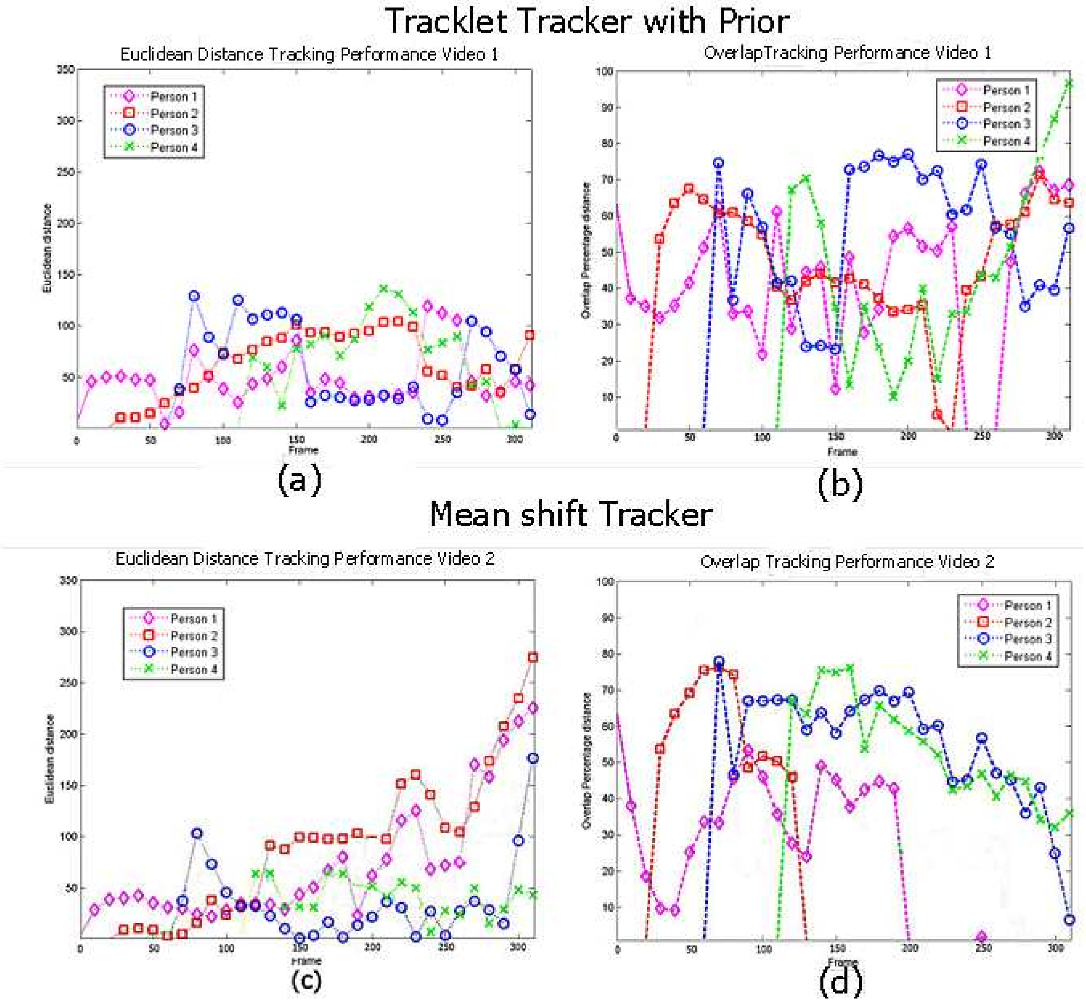


Figure 5.14: **Video 1:** Figure (a), shows the Euclidean distance between the computed trajectories and the groundtruth over the a video sequence(less is better) for the Tracklet Tracker with Prior. Figure (b), shows the percentage overlap between the computed Tracklet Tracker with Prior trajectories bounding box and the groundtruth's bounding box over the video sequence (more is better). Figure (c) shows the Euclidean distance between a Mean Shift tracker and the groundtruth. Figure (d) shows the percentage overlap between a Mean Shift tracker box and the groundtruth's box.

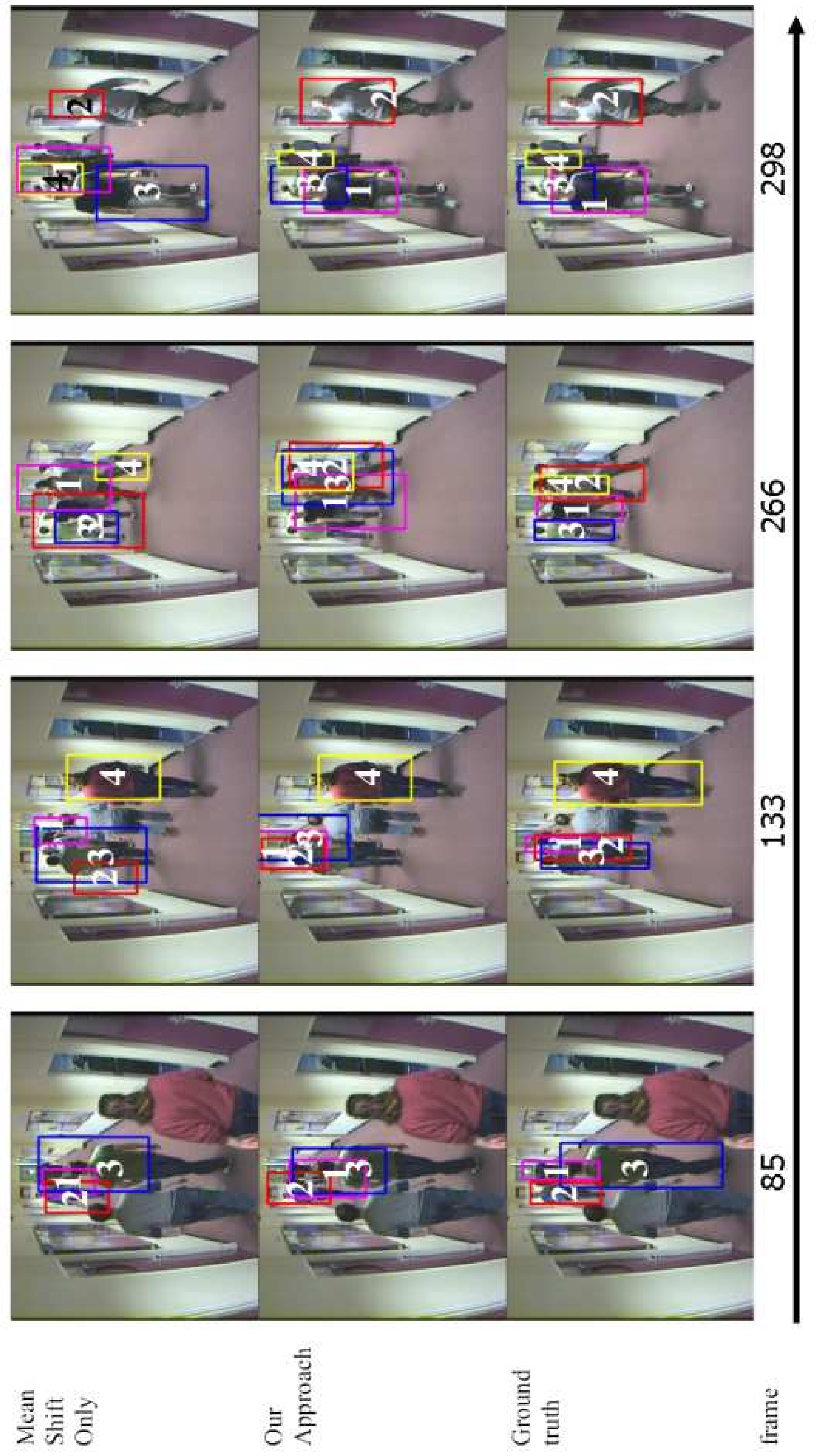


Figure 5.15: **Video 1:** A comparison between the tracklet tracker algorithm and a Mean Shift Tracker and the groundtruth over 4 frames.

qualitative comparison of 4 frames in the same video sequence. While the Mean Shift works well at frame 85, at Frame 153 there is full occlusion of a number of people being tracked and by frame 298 the Mean Shift has incorrectly tracked a number of people and this is shown by the increasing Euclidean distance and incorrect labelling of people in Figure 5.16. Figure 5.17 shows the mean per

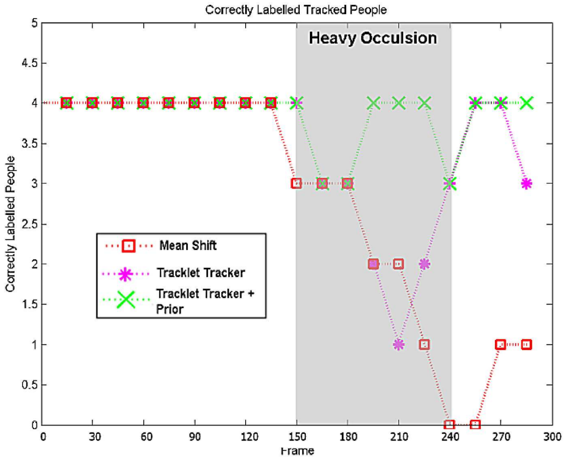


Figure 5.16: **Video 1:** To show the number of correctly labelled tracks (out of 4) over the course of the sequence.

frame Euclidean and percentage overlap for video sequence 2, this is a less difficult sequence, and both techniques work well. Though the Tracklet Tracker with Prior performs better than Mean Shift Despite these good results for 4 people in the sequence in video 1, the algorithm still fails part way on the other people in the sequence, including an individual who is continually occluded for around 100 frames by another person and his best path is corrupted. There are other people in the sequence who do not have trajectories formed for them, this is due to two main reasons. For people walking away from the camera, despite the head and shoulder detector working well, it isn't as reliable as people walking towards the camera. This means there are less tracklets to form the trajectory of the chosen people and the overall best path trajectory doesn't exist in the states. As

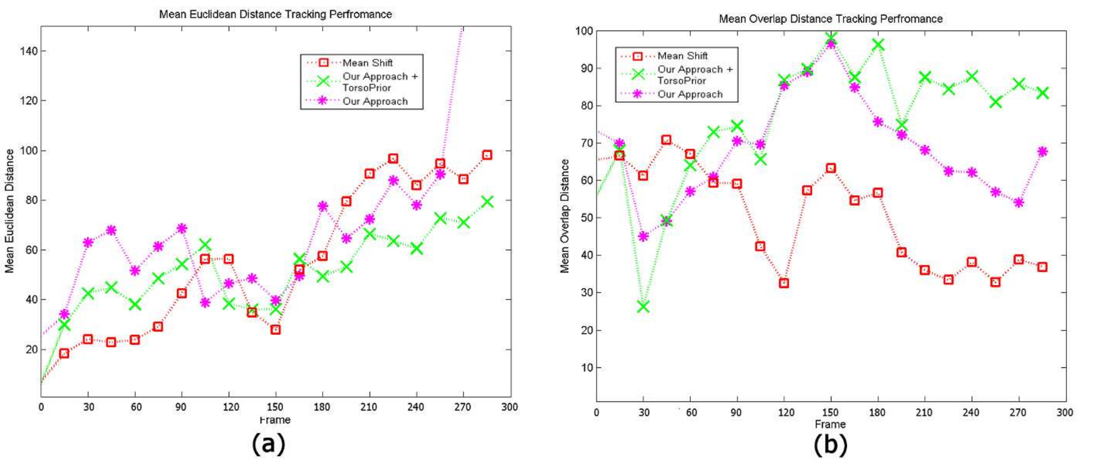


Figure 5.17: **Video 2:** A comparison of the mean Euclidean distance difference (Figure (a) , less is better) and percentage overlap (Figure (b) more is better) between the tracklet tracker algorithm and a Mean Shift tracker.

short but significant tracklets are linked to form a single trajectory through the sequence, if some of the people are continually occluded for long periods through the sequence, there will be no significant tracklet to link together and this will cause the tracking to fail.

5.6.3 CAVIAR Sequences

The CAVIAR sequence has a number of different challenges the techniques must cope with. The largest difference to the previous dataset *Dataset 1* video is the size of the people. The CAVIAR camera is mounted on the ceiling meaning people are smaller in size. However, a high mounting high means there is less occlusion due to an improved viewing angle. **Video 3** is taken from the sequence labelled “OneStopMoreEnter1”. Figure 5.18 shows that the Mean Shift is unable to track the selected people in the sequence, while the Tracklet Tracker has a high degree of success with a low euclidian error and a high mean overlap with the groundtruth. However there are times when the overlap drops significantly,

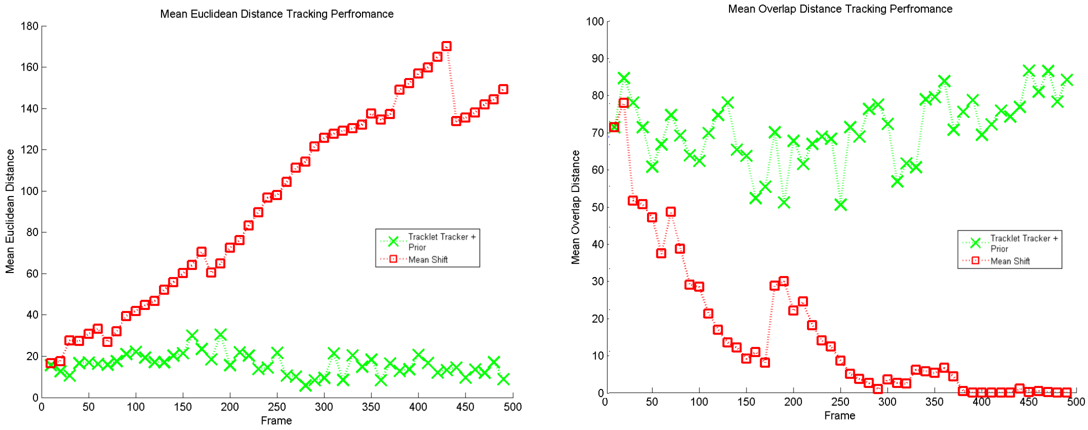


Figure 5.18: **Video 3:** A comparison of the mean Euclidean distance difference (Figure (a) , less is better) and percentage overlap (Figure (b) more is better) between the Tracklet tracklet with head and shoulder prior and a Mean Shift tracker.

(frames 150 and 250). This is caused by occlusions from other people, causing the tracker to become less accurately centred on the selected trajectory. The final sequence, **Video 4** from the CAVIAR dataset is shown in Figure 5.19 and it has a similar performance to Video 3. The Tracklet Tracker with head and shoulder prior has a number of jumps, which are caused by the piecewise linear nature of the tracklet tracker. These occur as a Mean Shift tracklet drifts off target, or if the human has a large change in motion, causing the motion model to briefly correct itself. To remove a number of these spikes the optical flow of the tracked person can be used to improve accuracy.

5.6.4 Improving Accuracy

To remove the spikes shown in Figures 5.18 and 5.19, the learnt optical flow prior is used to improve the bounding box accuracy. Figure 5.20 shows how the pixels are assigned in this example of minor occlusion from the CAVIAR dataset. It can be seen that despite the blue box overlapping the yellow area, few of the

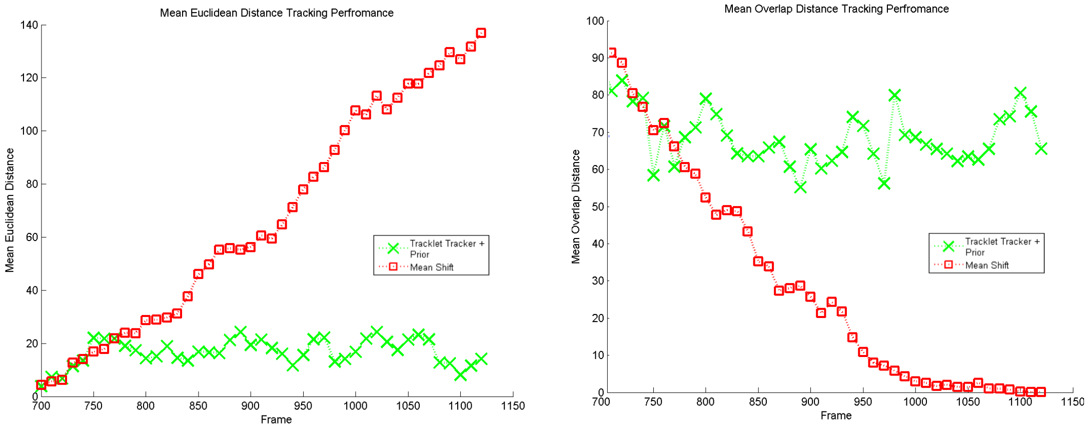


Figure 5.19: **Video 4:** A comparison of the mean Euclidean distance difference (Figure (a) , less is better) and percentage overlap (Figure (b) more is better) between the Tracklet tracklet with head and shoulder prior and a Mean shift tracker.

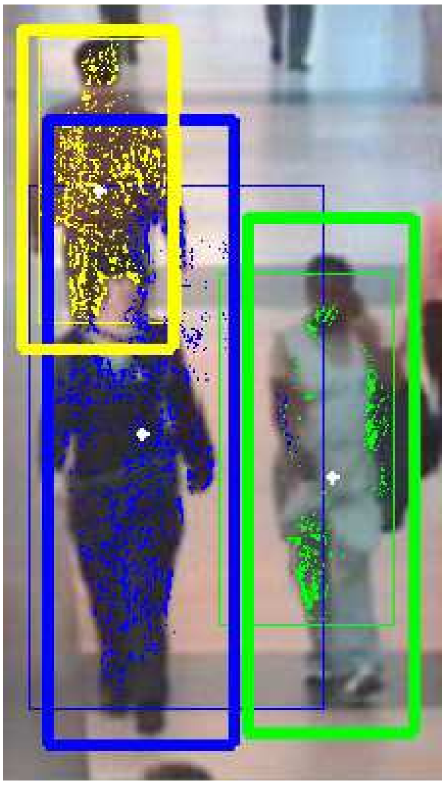


Figure 5.20: Example of assigned pixels when people have overlapping kernels.

pixels within the yellow area have been assigned as blue. The mean and standard deviation of the assigned pixels are found, to adjust the original object's bounding box. In Figure 5.20, the original blue box position and size is indicated by the thin blue line extending to the green person, this was then reduced in size to simply cover the correct person. This visible improvement can be seen in the performance graphs also. Figure 5.21 shows the improvement for **Video 3**. The mean euclidian distance error in Figure 5.21(a) for the Tracklet Tracker with prior *and* Optical Flow, is less than both the Mean Shift and Tracklet Tracker without the optical Flow priors. By using the Optical flow prior large increases in the overlap error shown by the sharp drops are smoothed out. This is due to the kernel centroid being adjusted to better fit the actual person being tracked. In addition, the error during the significant overlap that occurs at frame 170, is reduced for both the distance (Figure 5.21(a)) and kernel overlap (Figure 5.21(b)).

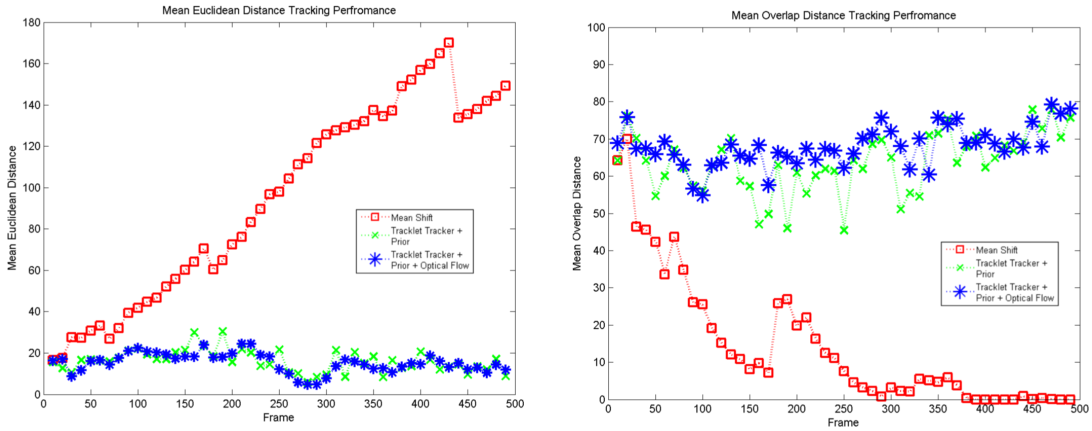


Figure 5.21: **Video 3:** A comparison of the mean Euclidean distance difference (Figure (a) , less is better) and percentage overlap (Figure (b) more is better) between the Tracklet tracklet with head and shoulder prior and the Tracklet tracklet with head and shoulder *and* optical flow prior.

A similar effect is seen within **Video 4** when the optical flow prior is used. Figure 5.22 shows the mean distance error and percentage overlap with groundtruth over the video sequence. There is a marked improvement around frame 1000 to

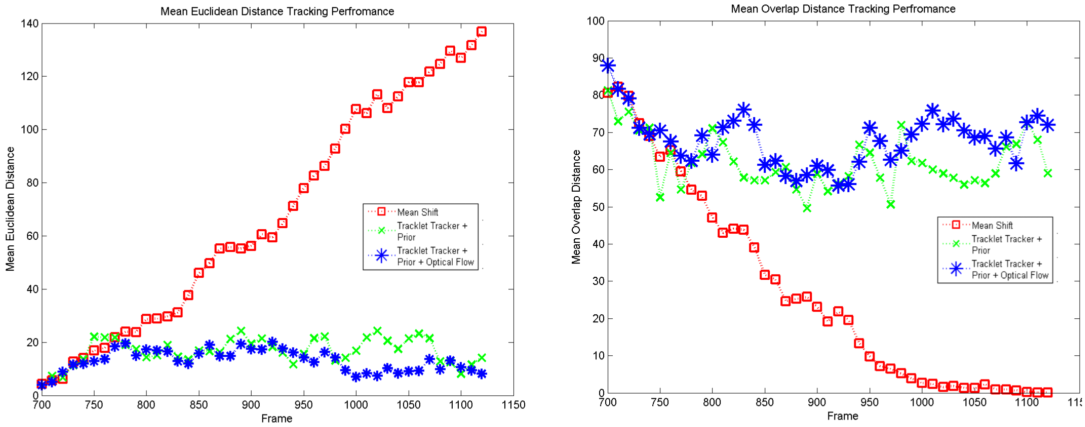


Figure 5.22: **Video 4:** A comparison of the mean Euclidean distance difference (Figure (a) , less is better) and percentage overlap (Figure (b) more is better) between the Tracklet tracklet with head and shoulder prior and the Tracklet tracklet with head and shoulder *and* optical flow prior.

1200 where using the Optical flow prior (shown as blue stars) significantly reduces the errors. Figure 5.23 allows the comparison of serval frames of images from Video 4's sequence.

It can be seen that at frames 915 and 1017, for the Tracklet Tracker with head and shoulder and optical flow priors, that the yellow centroid has been substantially adjusted to be more centred on the person.

5.7 Conclusion

This chapter has described an approach to combine the strengths of a global head and shoulder detector to locate people, with those of a localised Mean Shift tracker for frame to frame correspondence. The Mean Shift tracker produces short tracklets which are recombined using a Viterbi style approach to produce a full trajectory through the video. Two learnt models of human motion are

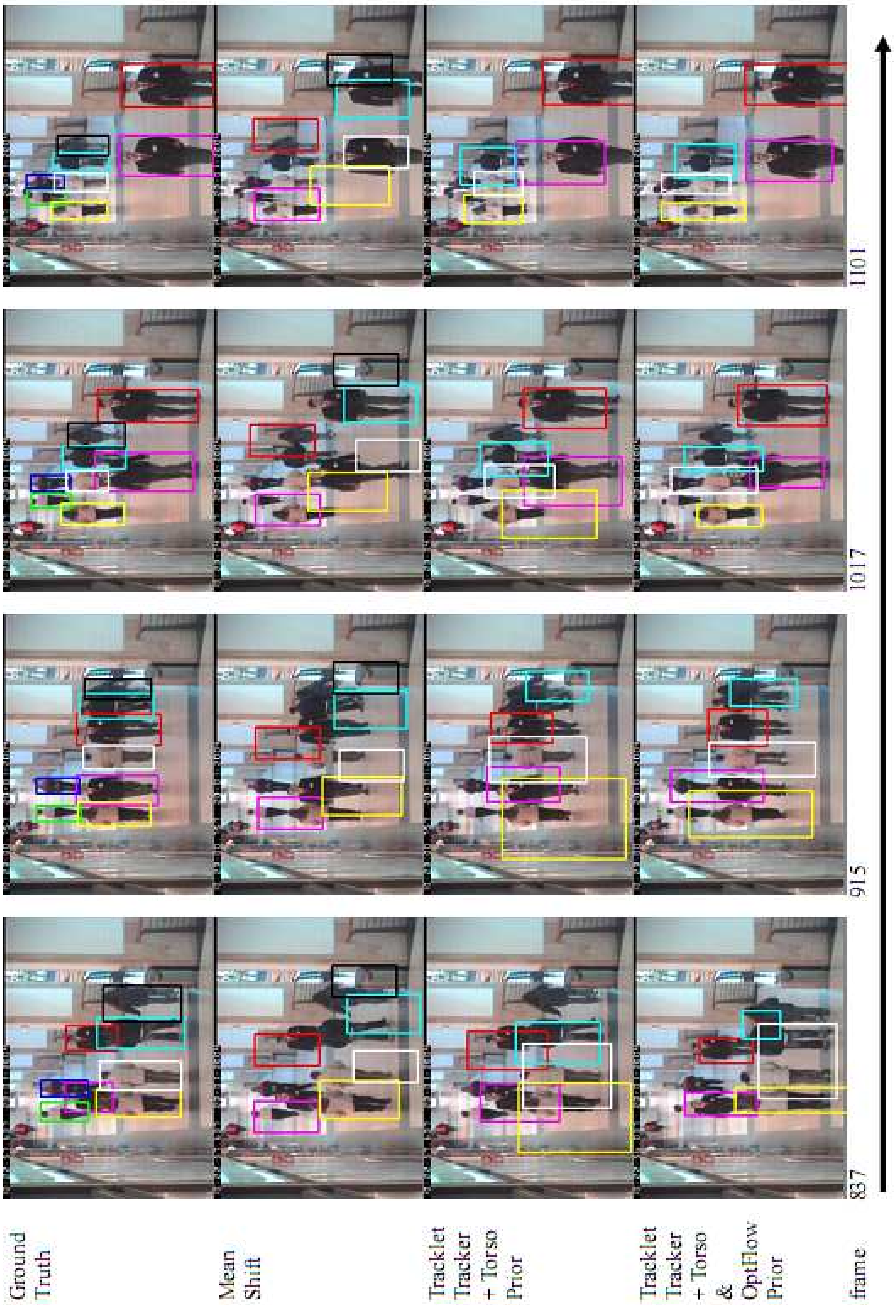

Ground Truth
Mean Shift
Tracklet Tracker + Torso Prior
Tracklet Tracker + Torso & OptFlow Prior
frame
837
915
1017
1101

Figure 5.23: **Video 4:** A selection of frames from the sequences, comparing all techniques.

added as constraints for the computed trajectory. The spatial reappearance of the head and shoulder detector over time has been used to model the motion of people, while learnt optical flow patterns of a persons kernel are used to improve accuracy of the trajectory. Overall this approach leads to an increased robustness to occlusions and interactions between people in crowded scenes.

The approach has been tested on four challenging sequences of people interacting and occluding within indoor scenes, including two on the well known dataset CAVIAR. The results are consistently better than that of a Mean Shift tracker which fails to cope with heavy occlusion. These promising results are possible using a single low cost surveillance type camera. Future performance could be further enhanced through detection of people facing away from the camera also by addressing the issue of people who are heavily occluded for long periods.

Chapter 6

Real Time Inter Camera Tracking

This chapter describes work into the tracking of objects between spatially separated uncalibrated cameras in real time. The transfer of a tracked object from one camera to another, can be termed "object handover". To be able to achieve successful object handover we need to know about the environment in which the cameras operate to be able to infer information about how objects move inter camera (between camera).

To address real world requirements, no *a priori* information is supplied to the techniques i.e. no colour, spatial or environmental calibration. As cameras may have no overlapping fields of view, many traditional calibration techniques are impossible. An ideal tracking environment could be described by the following:

- It is able to work immediately upon initialisation,
- Performance will improve as new evidence becomes available,
- Is adaptable to changes in the camera's environment

To be able to fulfil these aims the approach needs to learn the relationships between the non-overlapping cameras automatically. This is achieved by the way of three cues, modelling colour calibration, relative size and movement of objects inter camera. These are explained in sections; 6.5, 6.4 and 6.3 respectively. The three cues are deliberately weak as more detailed and complex cues would not be able to work with the low resolution and real time requirements of a typical camera installation. These three weak cues, are then fused together to allow the technique to determine if objects have been previously tracked on another camera or are new object instances. The approach learns these camera relationships, though unlike previous work does not require *a priori* calibration or explicit training periods. Incrementally learning the cues over time allows for the accuracy to increase without any supervised input.

This chapter presents a novel approach to inter camera tracking which fuses additional features with a scalable architecture providing accurate object handover between cameras. Unlike other methods this chapter presents work that is learnt incrementally over time instead of as a batch approach. The performance is demonstrated with extensive experimental testing and results and the incremental learning approach is compared with a traditional batch approach.

6.1 Experimental Setup

Figure 6.1 gives a general overview of the experimental setup. This figure shows an example containing two camera modules and a module for the operator to query the cameras about objects. Each camera is a self contained module connected to others via a network, meaning that it can easily be distributed over multiple processors or machines. An overview of each stage is given below:

- **Object Detection** The camera image is fed into an object detection module where a background scene model is maintained and updated. This model is used to delineate foreground from background for the incoming image.

- **Intra Camera Tracking** Foreground objects are correlated to objects in the previous frame using a Kalman filter to provide intra camera object tracking. If a correlation with an object in the previous frame is found, the object is labelled as an *Old Object* and the colour descriptor for that object is updated. If no correlation exists, it is labelled as a *New Object* and if an object from the previous frame has no correlation to any object in the current frame, it is deemed an *Exiting Object*. The Kalman filter continues positional predictions for the *Exiting Object* to overcome object occlusions, but after a set time with no incoming correlation, it is deemed to have left the camera. At this point, the *Exiting Object's* colour, size, and position descriptor are broadcast via the network to all other camera modules to enable inter camera tracking.

- **Inter Camera Tracking** When an object is labelled a *New Object*, its descriptor is compared to objects that have previously exited other cameras and been broadcast as potential candidates for object handover. This comparison is based upon colour similarity weighted by the prior of how colour varies between cameras and how likely the disappearance /reappearnce is..

- **Update System Cues** If a potential object handover is identified, the object's colour similarity is used to provide a weighted update to camera colour calibration model, the relative size of the bounding box, and iterative camera region linking scheme.

In this way each camera maintains a model of how other cameras in the network relate to it. To track an object through the camera network, a request queries all cameras, and possible correlations from the Inter camera Tracking cues are returned to the operator in a ranked list for the operator to use.

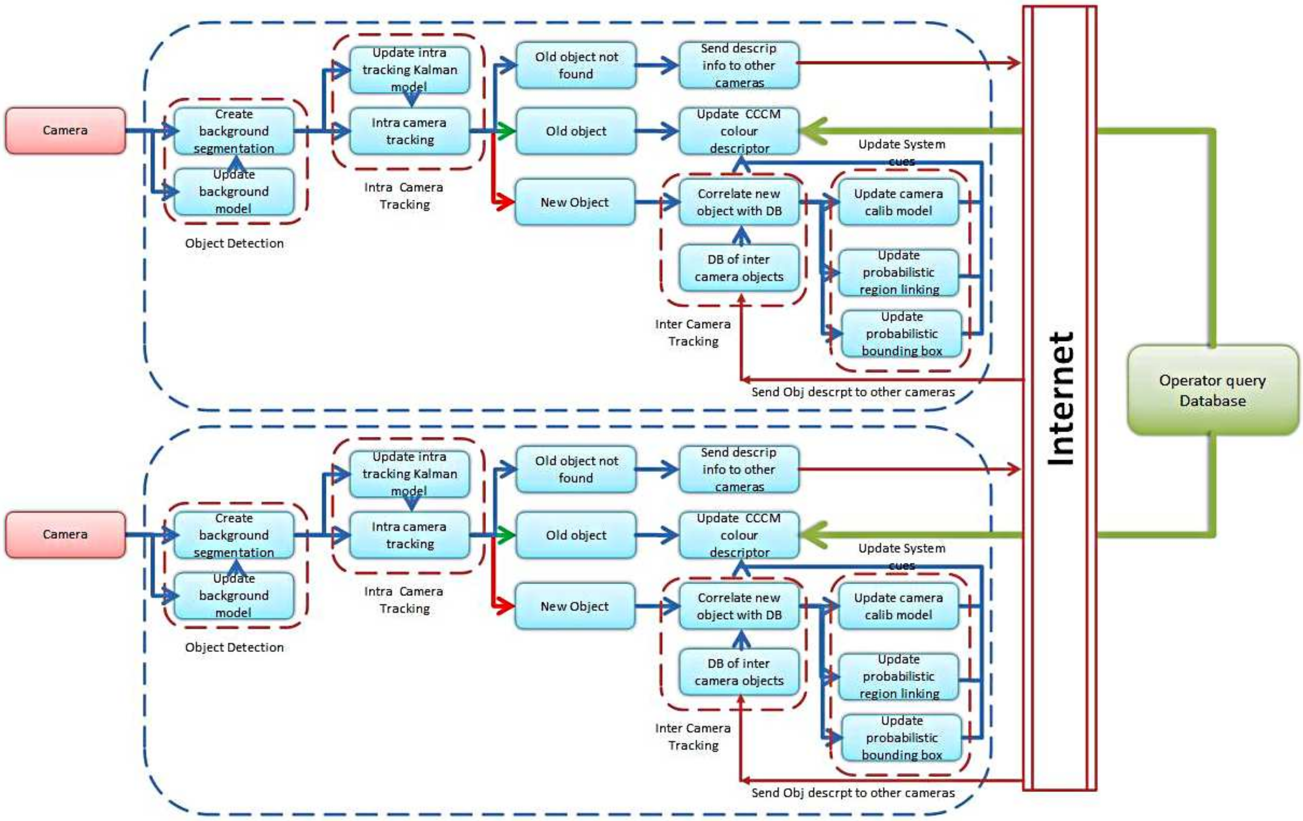

Camera
Create background segmentation
Update background model
Object Detection
Update intra tracking Kalman model
Intra camera tracking
Intra Camera Tracking
Old object not found
Old object
New Object
Send descrip info to other cameras
Update CCCM colour descriptor
Update System cues
Correlate new object with DB
DB of inter camera objects
Inter Camera Tracking
Update camera calib model
Update probabilistic region linking
Update probabilistic bounding box
Send Obj descrpt to other cameras
Internet
Operator query Database

Figure 6.1: System Overview with two independent camera tracking modules connected together by a network. With an operator query module for tracking a specific object inter camera

6.2 Intra Camera Object Tracking and Description

To detect moving objects, the static background is modelled in a similar fashion to that originally presented by Stauffer and Grimson [100]. The foreground objects are found from the background mask by connected component analysis on the resulting binary segmentation. This provides a bounding kernel around each object with a centroid and limits of the kernel. Further detail about the background modelling process is described in Section 2.2. This provides frame by frame object position, a track of each object through the sequence needs to be created using correlation between frames. This could be solved using the method in Chapter 5, where the technique combination of the head and shoulder detector and tracker gives high quality accurate results within challenging scenes. However that work is an offline process due to head and shoulder detector being computationally complex. Due to the large data collation phase of this approach where the camera relationships are learnt, a real time approach is required. Therefore to reduce the computational demand for the intra camera tracker a Kalman Filter is used to provide good temporal linkage between the detected foreground objects. This will reduce the robustness of tracking individuals within crowds, but will still maintain the ability to track a crowd as one group. Therefore will not reduce inter camera robustness.

6.2.1 Kalman Filter

A Kalman filter is used to correlate the inter frame movement of foreground objects. The filter is a recursive process in that each updated estimate of the state position is computed from the previous estimate and the new input data, so only the previous estimate is kept. In addition to eliminating the need for storing the entire past observed data, the Kalman filter is computationally more efficient than computing the estimate directly from the entire past observed data at each step of the iterating process. This means it can be run in real time, while having the ability to use the predictive estimates to allow for minor occlusions. In addition, the trajectory of the object is smoothed despite minor jumps caused by the connected component analysis of the background mask. Each new object has a Kalman filter initialised on its position and is updated using the standard Kalman filter update rules [109] using a constant velocity model with a white noise drift term. For details Section 3.3 in Chapter 3 provides detail about the update equations. The Kalman filter estimates are represented by the mean and covariance of the centre of an object's bounding box. These are correlated to the detected foreground objects using Mahalanobis distance.

6.2.2 Object Appearance Modelling

The appearance of a foreground object within a kernel bounding box is represented by a discrete histogram. Each foreground pixel within the kernel is quantised to a bin within the appearance histogram. A histogram is used as the appearance descriptor for the objects as, it is invariant to pose and shape, making it ideal for cross camera correlation. Chapter 4 explored the idea of using colour as a descriptor for both intra and inter camera tracking of people. Within that chapter the colour consistency of object tracked inter and intra camera were investigated with a number of techniques analysed. A brief summary of the findings are provided below, The Colour Lookup Table colour space (CLUT) quantisation model will be used initially as this has a high performance tracking objects. This method works well with the real time constraints (25fps) with uncalibrated cameras, providing a simple coarse descriptor for both intra and inter camera tracking of objects. However, as described later, one of the learnt cues is the linear colour transformation of objects inter camera. This will allow for basic colour calibration to occur. For this, a linear colour space such as RGB must be used. Therefore once the initial stage of learning the colour model inter camera has occurred, RGB quantisation with a computational cost effective Post-Parzen window is used. This provides a higher performance as it will be partly calibrated allowing a discriminative object correlation.

6.2.3 Camera fields of view

The location and environment of the surveillance cameras will determine whether cameras have overlapping or non-overlapping fields of view. In real case situations such as airports and rail stations, most cameras will be non-overlapping as the number of cameras is limited by physical and cost constraints. When cameras have overlapping fields of view, correlation between the cameras needs to be learnt, to be able to handover object tracking as objects move between the cameras. This is traditionally done through explicit manual geometric calibration of the system with a known object, for example a colour chart or checker board. However, there is no calibration between overlapping cameras within this work, instead the cameras will learn over time that as one object appears on camera 1 and then appears on camera 2 a strong temporal relationship between the two cameras exists. Cameras with non-overlapping fields of view pose a challenging problem to object handover, as the objects may never be observed simultaneously. However, the same method can be used for overlapping and non-overlapping cases.

There are a total of eight cameras used to test the techniques. They are spread over two floors, in two close groups of four cameras. Figure 6.2 shows the two groups of cameras with the physical link shown by the arrow. The lower floor contains a large number of popular alternative exits which allows people to use multiple routes. In addition camera 8 faces a lift and door which create foreground motion in addition to the actual objects of interest. Simple quantisation of colour alone cannot correlate objects sufficiently inter camera due to illumination variations, and occlusions. Therefore, three methods to concurrently learn the relationships between the cameras are presented in Sections 6.3 to 6.5.

6.3 Probabilistic inter camera coupling

The first method incrementally learns the probabilistic relationship of object movement between cameras. This makes use of the key assumption that, given time, objects (such as people or cars) will follow similar routes inter camera due to paths, shortest routes and obstructions. The repetition of these routes over

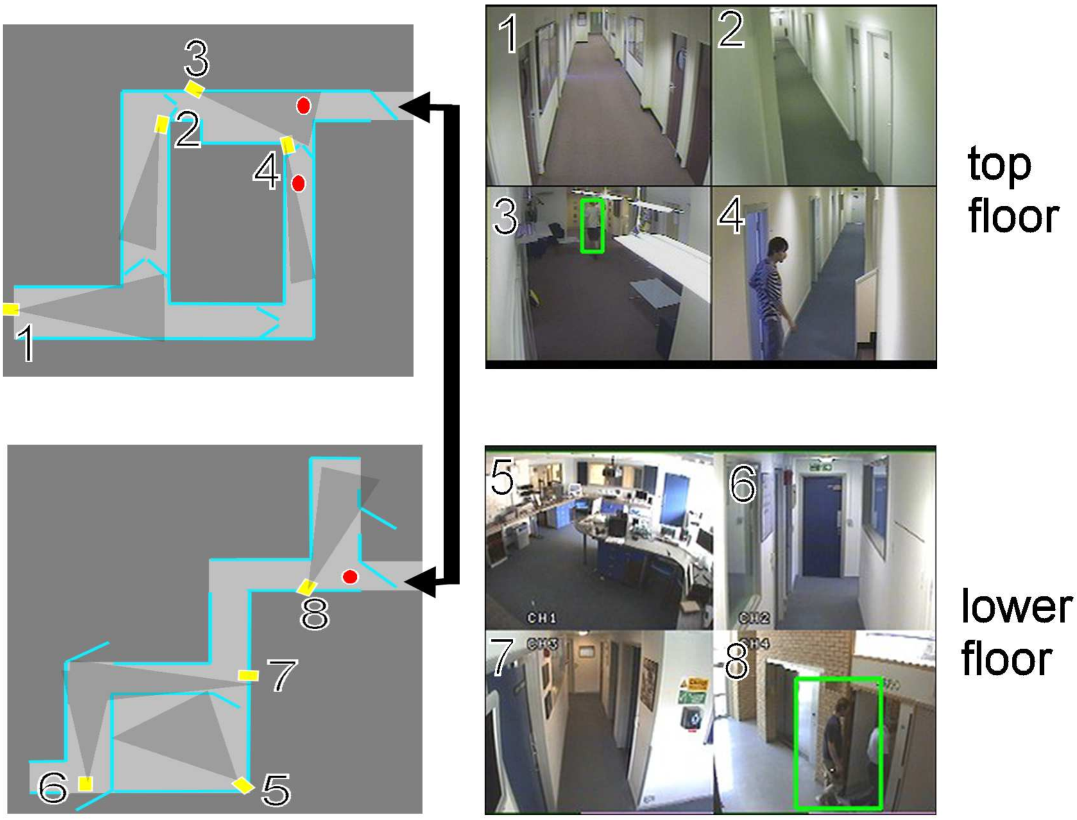


Figure 6.2: The two camera setup over the two floors with example images.

time, will start to form marked trends in the data. The reappearance period between two cameras is modelled by calculating the probability that an object disappearing from one camera at time t^{start}, will reappear in another camera at time t^{end}. These probabilistic temporal inter camera links can be used to couple camera regions together, producing a probabilistic distribution of an objects movement between cameras.

This makes it possible to link common entry and exit regions between cameras. The links, modelled as conditional probabilities, are constructed using reappearance histograms populated over time as evidence is gathered. As the number of possible links increase, so does the quantity of data required to populate the histograms. However, most links are invalid as they correspond to impossible routes, such as entry points on walls, or between cameras too distant to be reliable. Therefore, the technique is able to identify the valid and invalid links without user supervision. Previous solutions required either batch processing or hand labelling to identify entry/exit points, both impractical in large systems, and unable to adjust to camera or environmental changes. This approach is initially coarsely defined, but increases in detail over time as evidence becomes available, and can adjust to changes without a system restart.

6.3.1 Incremental link learning

Objects are automatically tracked intra camera with a Kalman filter to form a colour appearance model of the object. The CLUT colour histogram $B = (b_1, b_2....b_n)$ is the median histogram recorded for an object over its entire trajectory within a single camera. A median histogram is where each bin is found by taking the median value of that bin over the person's trajectory. All new objects that are detected are compared to previous objects exiting other cameras within a set time window, T. The image is split into a number of areas called regions for entry and exit of people. Between the regions a temporal link is formed. This temporal link is a a discrete probability distribution of an objects reappearance period T. It is formed using the colour correlation of new objects with respect to their recorded reappearance period. The colour correlation is computed using Bhattacharyya similarity measure. Other correlation measures such as histogram intersection and mutual information were examined in great detail Section 4. However, for correlation of object both intra and inter camera, the Bhattacharyya similarity measure give the highest performance. Thus the frequency f of a u bin in a temporal link reappearance model between two regions is calculated as

$$f_u = \sum_{\forall r} \sum_{\forall s} \begin{cases} \rho[r,s] & u\Delta i \leq (t_r^{end} - t_s^{start}) < (u+1)\Delta i \\ 0 & otherwise \end{cases} \qquad \forall u, u\Delta i < T \tag{6.1}$$

where t_r^{start} and t_r^{end} are the entry and exit times of object r respectively, T is the maximum allowable reappearance period and Δi is the bin size in seconds. $\rho[r,s]$, the Bhattacharyya similarity measure between the appearance models of objects r and s is calculated as

$$\rho[r,s] = \sum_{u=1}^{m} \sqrt{r_u s_u} \tag{6.2}$$

To reduce quantisation effects, a gaussian kernel blur can be used. Therefore For In-Parzen Windowing equation 6.1 becomes

$$f_u = \sum_{\forall r} \sum_{\forall s} \rho[r,s]\eta((t_r^{end} - t_s^{start} - u\Delta i), I\sigma^2) \qquad \forall u, u\Delta i < T \tag{6.3}$$

where $\eta(\overline{V}\epsilon\Re, I\sigma^2)$ represents a 1D Gaussian kernel positioned at V with covariance σ^2

$$\eta(\overline{V}\epsilon\Re, I\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{t^2}{2\sigma^2}} \tag{6.4}$$

Frequencies are only calculated for an object p that disappears from region $\beta 2$ followed by a reappearance in region $\beta 1$, $(f^{\beta 1|\beta 2})$. By normalising the total area of the histogram by $\sum_r^T f_\phi^{\beta1|\beta2}$, an estimate to the conditional transition probability $P(O_{x,t}|O_y)$ is obtained. An example of $P(O_{\beta1,t}|O_{\beta2})$ is shown in Figure 6.3 where $O_{\beta1,t}$ is object $\beta1$ at time t. This probability distribution shows a distinct peak at 9 seconds indicating a link between cameras 1 and 4 with a single region per camera.

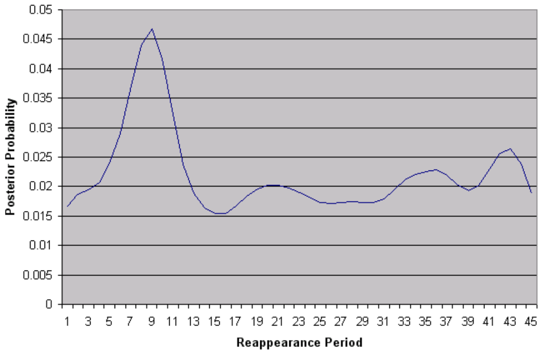


Figure 6.3: An example of a probability distribution showing a distinct link between cameras 1 and 4 with a single region per camera over a reappearance period of 45 seconds.

After sufficient evidence has been accumulated, determined by the degree of histogram population, the noise floor level is measured for each link. This could be determined statically using the mean or variance, however, through experimentation, using double the median of histogram values was found to provide consistent results. Figure 6.4 shows how the reappearance period of objects between cameras 3 and 2 of Figure 6.2 develops as observations are added over time. A peak reappearance probability at around 10 seconds increases in height as people are tracked and evidence is added to the distribution. After 1000 people have been accumulated there is a distinct peak around a reappearance period of 10 seconds.

If the maximum peak of the distribution is found to exceed the noise floor level, this indicates a possible link relationship between the regions. If a possible link

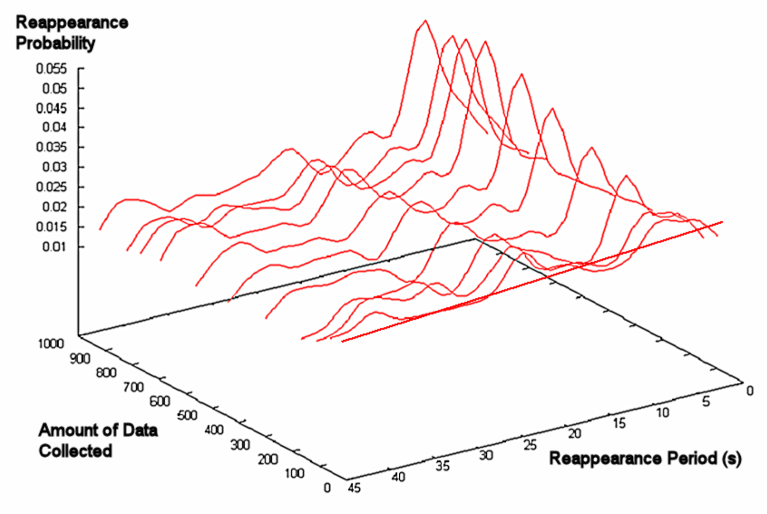


Figure 6.4: The reappearance period probability between camera 3 and 2 with increasing collected data up to 1 day.

has been found, the parent regions are subdivided into four child regions as in Figure 6.5. The initial distributions of the four new regions are set to that of the parent. Subsequent data is incorporated into the appropriate refined distribution.

In order to allow for multiple entry and exit areas on the cameras, each camera is subdivided into a number of equal regions, 16 on the current experiment. Coupling all regions to all others is only feasible in small-scale experimental approaches. As the number of cameras increase, the number of links required to model the prior will increase exponentially. With 16 regions between two cameras, there are 16^2 (256) links, with just an extra two cameras this becomes 16^4 (65536) links. However, many regions will not form coherent links, and can therefore during the subdivision operation, unused regions can be removed to minimise the number maintained. It is important that links are not removed between regions that simply require additional data. Therefore, a link between two regions is only removed if it has no data in it at all. This cautious method ensures no regions or links are removed that might be useful in a later subdivision. Figure 6.5 shows how the active regions are sub divided and removed over time. Initially there are no regions as shown in *initial start-up*, then each camera is assigned a single region, with a uniform conditional probability of objects moving between cameras. After the first 367 tracked objects, subdivision 1 shows how the cameras are linked together. Subdivision 2 occurs after 1372 objects have been added to the technique and unlinked regions are removed. Further subdivision and removal of regions is achieved in subdivision 3 following 2694 objects and subdivision 4 at 7854 objects. The remaining regions in subdivision 4 show the main entry and exit areas. Table 6.1 shows the number of links maintained and dropped at each subdivision stage, along with the amount of data used. It can be seen that with each iteration, the number of possible links increases dramatically, whereas the number of valid links kept is considerably less. The policy of removing unused and invalid regions improves the approaches scalability. This iterative process can be repeated to further increase the resolution of the blocks. The regions start to form the entry and exit points of the cameras, Figure 6.6a, shows the result after 4 subdivisions. The lighter regions have a higher importance determined by

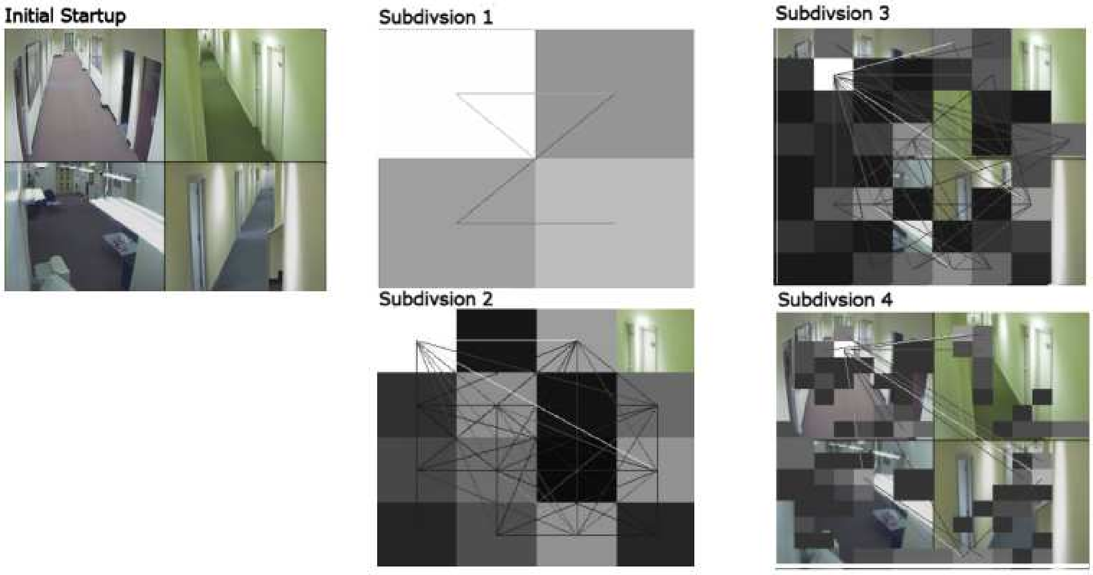


Figure 6.5: The iterative process of splitting the blocks on the video sequence over a day.

Iteration	Amount of Data	Number of Regions	Tot poss Links	Initial links	Dropped links	Kept links
1	367	4	12	12	0	12
2	1372	16	240	240	45	195
3	2694	60	2540	1631	688	943
4	7854	191	36290	36134	34440	1694

Table 6.1: Table of number of links maintained and dropped in each iteration of region subdivision.

the number of samples each link contains. As the number of iterations increase, the size of the linked regions decreases and thus reduces the number of samples detected in each region. This affects the overall reliability of the data used. To counter this, regions which are found to have similar distributions to neighbouring regions are combined together to increase the overall number of samples within the region (as illustrated in Figure 6.6b,). This reduces the overall number of

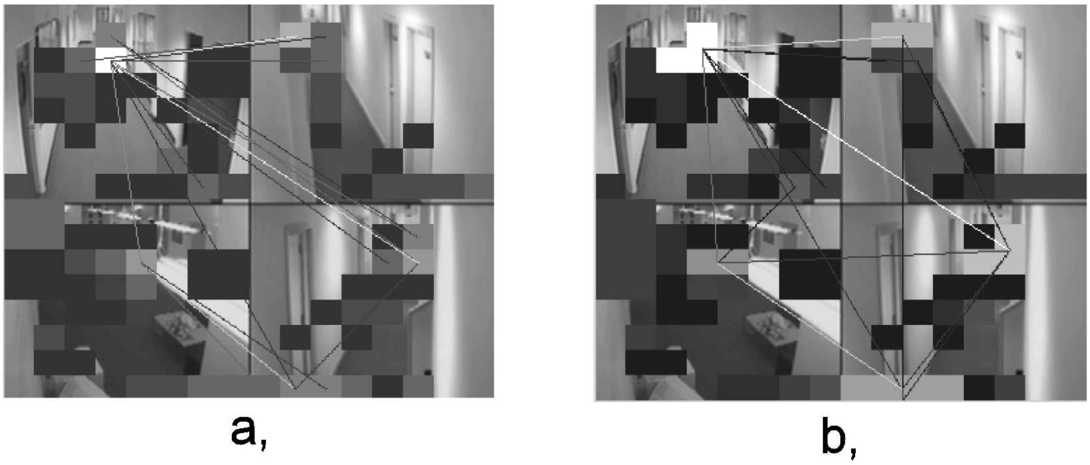


Figure 6.6: a, shows the main identified entry/ exit regions. b,shows the individual regions that, if similar, are then recombined to form larger better populated regions, shown by the constant colour areas.

regions maintained and the actual links between regions, therefore increasing the accuracy of the remaining links. This incremental approach of learning entry and exit areas, is similar in outcome to that of a batch approach. With a batch technique all data is first collected and then all data is concurrently used to compute the entry and exit points. This could produce a more accurate result, however it is only possible after the data collection phase is over.

6.4 Probabilistic Inter Camera Bounding Box

The background segmentation used, provides a background mask, with the foreground objects labelled via connected component analysis. Around each of the detected objects a bounding box is formed based on the mean and standard deviation of the blob in pixels, Figure 6.7 shows the rectangular bounding box of a person. The size of this bounding box is utilised to provide a coarse size descriptor of the object in the image plane as it moves upon the ground plane. Objects further from the camera will have a smaller bounding box size, with closer objects having a larger size. As in the previous section we can assume that, over time, objects follow similar routes inter camera. This means that they will exit and enter cameras in consistent areas and therefore the size of the object should

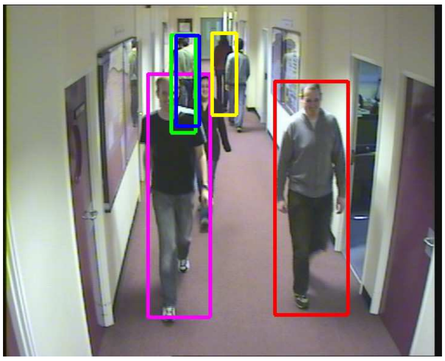


Figure 6.7: The rectangular bounding box around a detected object.

be consistent upon entry or exit. Looking at the experimental environment in Figure 6.2, if a person moves between camera 4 and 3, they will leave camera 4 at the bottom of the camera with a large bounding box, and should reappear towards the top of camera 3 with a relatively small bounding box. This fact can be utilised to calculate the likelihood that a person has moved inter camera based upon the relative entry size to the current camera.

The relationship between the exit size from one camera and the entry size in another can be represented by a 3D histogram, but due to the problems associated with having sufficient observations to populate such a histogram we assume independence and model the relationship as two 2D histograms. Figure 6.8 shows this relationship for the x size of the bounding box between cameras 1 and 4. These relationships are only modelled at a camera-to-camera level and for eight cameras there are 56 discrete histograms. The 2D distributions are learnt over time in a similar way to the probabilistic spatio temporal coupling in the previous section.

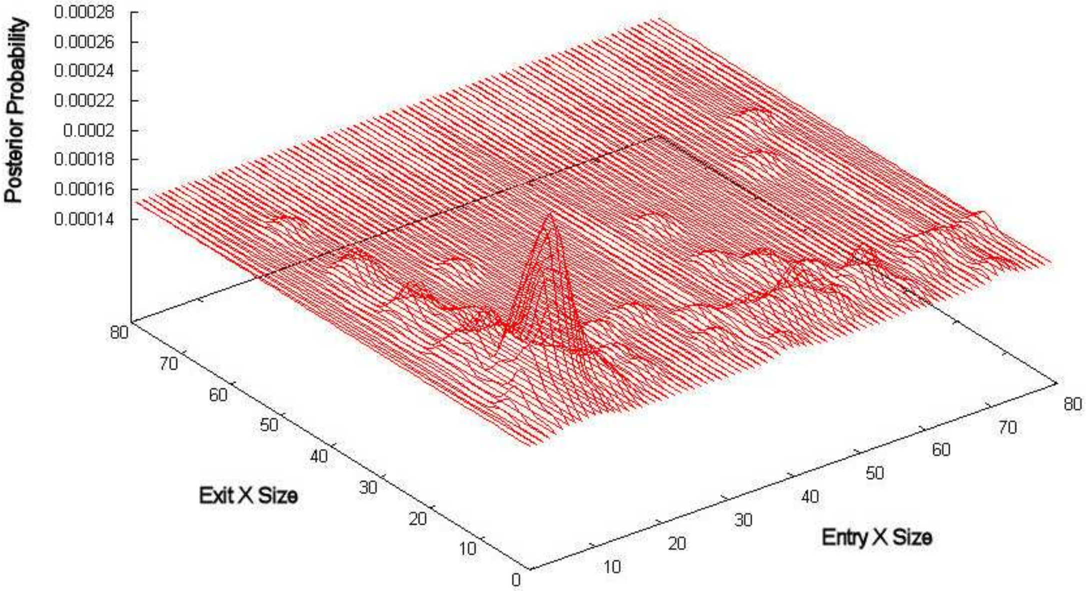


Figure 6.8: The probability distribution of the horizontal bounding box between cameras 1 and 4.

The histograms all start uniformly distributed. All new objects are compared to previous objects within the reappearance period T. The correlation is computed using the Bhattacharyya coefficient measure in equation 6.2. The new object s will have an entry size and the old object r an exit size, so the observation is then used to increment the appropriate bin in the 2D histogram for the specific link by the strength of the correlation (colour similarity). Thus the frequency of a bin for the x axis between two cameras $cam1$ and $cam2$ with the bounding box, $size$, is calculated as

$$f(size_{cam1}^{exit}, size_{cam2}^{entry}) = \sum_{\forall r} \sum_{\forall s} \rho[r,s] * \eta((size_r^{exit}, size_s^{entry}), I\sigma^2) \;\; (t_r^{end} - t_s^{start}) < T \tag{6.5}$$

where $\rho[r,s]$ is the Bhattacharyya similarity measure between the objects from equation 6.2 and $\eta(\overline{V}\epsilon\Re^2, I\sigma^2)$ is a 2D Gaussian as

$$\eta(\overline{V}\epsilon\Re^2, I\sigma^2) = \frac{1}{2\pi\sigma^2\sigma^2} e^{-[\frac{(x-\mu_x)^2}{2\sigma^2} + \frac{(y-\mu_y)^2}{2\sigma^2}]} \tag{6.6}$$

$f(size_{cam1}^{exit}, size_{cam2}^{entry})$ is then normalised, by the total area of the distribution to provide the conditional probability of the resulting change in bounding box size $P(O_{Entry}|O_{Exit})$ where O_{Entry} is an object with an entry size of $Entry$ and O_{Exit} is a previous object with a size of $Exit$. Over time, the prior of the size of entry and exit bounding box will become increasingly accurate as more data is collected. This is then used to weight the observation likelihood obtained through colour similarity as was done in the previous section. As will be seen in Section 6.8, these cues can be combined by simply multiplying their likelihoods together.

6.5 Inter Camera Colour Calibration

Colour quantisation assumes a similar colour response between cameras. However this is seldom the case, the cameras of Figure 6.2 show a marked difference in colour response even to the human eye. Therefore, a colour calibration of these cameras is proposed that can be learnt incrementally as with the distributions previously discussed.

Initially the CLUT colour descriptor is used as the correlation measure between objects. However, once sufficient colour calibration is achieved, a traditional RGB quantisation with Post-Parzen Windowing is used as this provides a greater level of discriminative detail. The colour transformation matrices between cameras are constructed in parallel with the construction of priors on reappearance probability and size. The tracked people are automatically used as the calibration objects, and as shown in Figure 6.9, a transformation matrix is formed incrementally to model the colour changes between cameras. As people vary in size, a point to point transformation is unavailable. We therefore use the colour descriptor (a histogram) of the object in the different cameras to provide the calibration. As the histograms represent the probability distribution of an objects colour within a camera, a linear transform is capable of providing the histogram pdf descriptor to histogram pdf descriptor mapping [31] between cameras.

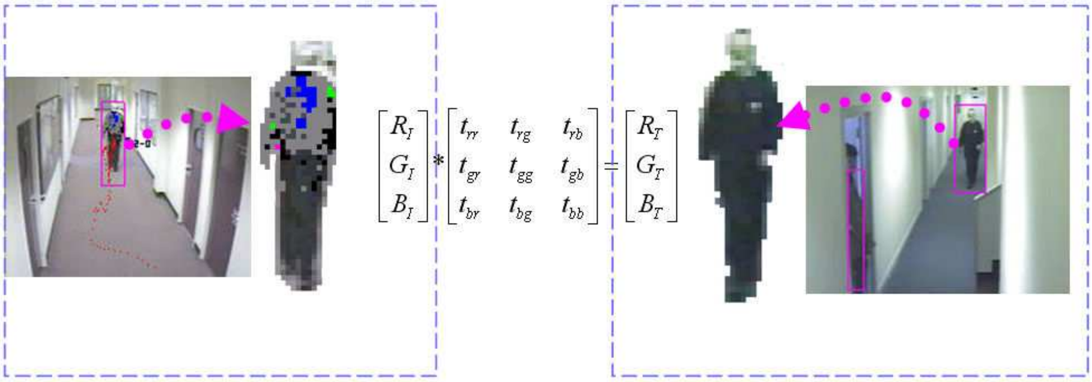

$\begin{bmatrix} R_I \\ G_I \\ B_I \end{bmatrix} * \begin{bmatrix} t_{rr} & t_{rg} & t_{rb} \\ t_{gr} & t_{gg} & t_{gb} \\ t_{br} & t_{bg} & t_{bb} \end{bmatrix} = \begin{bmatrix} R_T \\ G_T \\ B_T \end{bmatrix}$

Figure 6.9: An illustration of when the transform matrix is used.

Transformation matrices are formed between the four cameras. Six transformations along with their inverses provide the twelve transformations required to transform objects between the four cameras. As camera calibration is refined, the illumination changes that affected the success of the original correlation methods discussed in [15] and Section 4, are reduced. This allows the object descriptor used to be changed from the coarse CLUT to an RGB quantisation, which is more discriminative.

The six transformation matrices for the four cameras are initialised as identity matrices assuming a uniform colour response between cameras. When a person is tracked inter camera and identified as possibly the same object, the transformation between the two colour descriptors is calculated, $R * H = S$. Where the transformation matrix H is calculated by computing the transformation that maps the person's descriptor from the previous camera R to the person's current descriptor S. This transformation is computed via SVD and each matrix element is weighted by the objects colour similarity between the two colour descriptors. This weighting is essentially learning rate, allowing strong correlation to be heavily incorporated into the colour calibration models, while uncertain correlations given less important. The matrix t is then averaged with the appropriate camera transformation matrix, and repeated as people are tracked between cameras to gradually build a colour transformation between the cameras. As not all object correspondences will be true correspondences, this method will introduce small errors. However, it is in keeping with the incremental theme of the thesis and again relies upon the fact that given time, statistical trends in the data will emerge. This allows continual updating and adapting to the colour changes between cameras as additional data becomes available.

6.6 Calculating Posterior Appearance Distributions

The conditional prior probability cues of objects between cameras can be used to weight the observation of tracked people providing a posterior probability that an object has been tracked inter camera. Over time, the prior and therefore the posterior becomes increasingly accurate as available data increases, this allows for region subdivision, and increasingly accurate colour calibration and bounding box priors. Given an object O_t which disappears in region $\beta 2$ we can model its reappearance probability over time as;

$$P(O_t|O_{\beta 2}) = \sum_{\forall \beta 1} w_{\beta 1} P(O_{\beta 1,t}|O_{\beta 2}) \tag{6.7}$$

where the weight $w_{\beta 1}$ in region $\beta 1$ at time t is given as

$$w_{\beta 1} = \frac{\sum_{i=0}^{T} f_i^{\beta 1|\beta 2}}{\sum_{\forall \beta 2} \sum_{i=0}^{T} f_i^{\beta 1|\beta 2}} \tag{6.8}$$

This probability is then used to weight the observation likelihood obtained through colour similarity to obtain a posterior probability of a match, across spatially separated cameras. Bayes provides a method to estimate the posterior.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{6.9}$$

where $P(B|A)$ is the prior conditional probability $P(O_t|O_{\beta 2})$ from equation 6.7 and $P(A)$ the observation likelihood $H_{\beta 1 \beta 2}$. Thus the posterior for a newly detected object $\beta 1$ being object $\beta 2$ at time t can be given by

$$P(O_{\beta 2}|O_{\beta 1}) = observation * prior = P(O_{\beta 1,t}|O_{\beta 2}) * H_{\beta 1 \beta 2} \tag{6.10}$$

Tracking of objects is then achieved by maximising the posterior probability within a set time window.

6.7 Scalable Framework

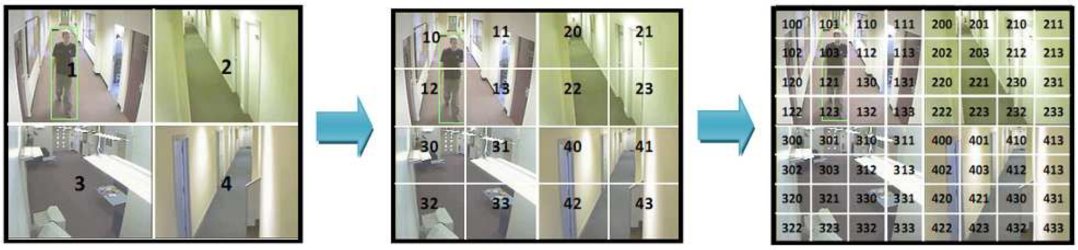
Within a large network it is not feasible to allow every sensor to communicate directly with every other sensor; the results of which would swamp the physical network with irrelevant information and waste valuable memory and processing time. This section discusses design considerations for large surveillance networks and how the proposed method is scalable. As the number of cameras increase in the network, the architecture of the network and data communication between modules become an important consideration. Traditional systems are based on a client server architecture. With the server receiving and processing all the video feeds. Communication between cameras is then carried out within the server ensuring high speed. However, as all processing is performed by the single core server, should the server fail, the whole network would be immobilised. In addition, the network would be limited by the processing speed of the server, and adding further cameras, would slow the overall performance.

An alternative which this technique utilises is based is a decentralised network which operates as a Peer-to-Peer network. In a Peer-to-Peer architecture, there are no servers or clients, but only equal *peer* nodes that function simultaneously as both “clients” and “servers” to the other camera nodes on the network.

6.7.1 Scalable Learning and Tracking

The main bottleneck of a peer-to-peer implementation is the increased bandwidth requirements between camera nodes. Therefore, the minimum amount of communication between cameras is essential. When a person is detected on a camera, they will be tracked within the camera while visible. As the object exits the camera, their descriptor and the leaving position is broadcast to all other cameras to be stored locally. As each camera doesn’t record the level of region subdivision at the destination of its links, a formalised labelling of the region link is used.

The technique is based on a rectangular region subdivision where each camera is divided into 16 regions. To allow for a scalable technique, each region has a 4 digit number which corresponds to the level of subdivision the region has undergone and its originating camera. Initially there is one region per camera, this allows immediate tracking with the links initially uniformly distributed. Figure 6.10 shows how the subdivision takes place for camera X, with the region ID adding an additional digit for each subdivision. At the first level of subdivision, a single digit is used, then when subdivided another digit is added (Figure 6.10b). This means that the complete ID for each region also contains the ID for the higher level regions, i.e. the ID X113, says that region 113 is part of the region 11 at a higher level which in turn is part of the region 1 on camera X. This means that links between two cameras can be constructed at the highest resolution support by both cameras. In the example of Figure 6.11, a person has just left camera 1, allowing camera 1 to broadcast their descriptor, along with the exit region which is 13, making the region ID 113 to all other cameras. A new person appears on camera 2 and will use the descriptor from 113 to the new region (211) on its camera as required. However, when another person is detected on Camera 3, the region on camera 3 has a link to camera 1 at a less detailed level due to lack of data. Therefore camera 3 will use the link between region 1 on camera 1 and



(c) Level 1 Regions (b) Level 2 Regions (a) Level 3 Regions

Figure 6.10: The First3 levels of region subdivision with their associated numbering. The star indicates a highlighted region and its ID below. (a) shows the initial camera regions, (b) after 1 sub division and (c) after two subdivisions.

region 2 on camera 3. Both cameras 2, and 3 use the same formalised region ID despite them having links to camera 1 that were at varying levels of detail.

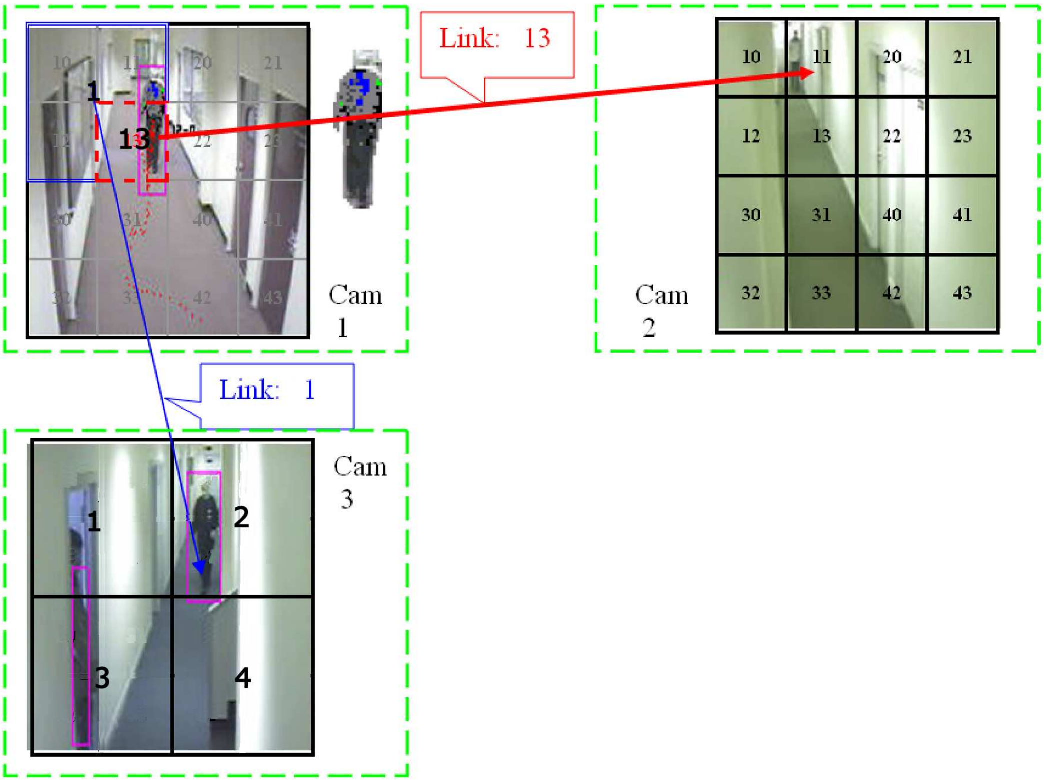


Figure 6.11: Example of the scalable region linking based around a formalised region ID system with the person in camera 1 linked to cameras 2 and 3. With a detailed link to camera 2 from region 13 to region 11, and a less detailed link to camera 3 from region 1 to region 22.

For an operator to track a specific object, the operator communicates with the camera the object is currently on, setting a tracking flag. Then as this objects leaves the camera, it is broadcast with the tracking flag. When another camera finds a correlation to the flagged object, the operator is informed of the event without further communication to the original camera. Other cameras that find further correlation within the time threshold also inform the operator about the matches. This allows the operator to make a decision from the top ranked matches returned.

6.8 Experimental Results

This section demonstrates the performance of the techniques proposed for tracking objects across up to eight uncalibrated overlapping or non overlapping cameras.

6.8.1 Experiment Setup

The experimental setup consists of eight colour cameras with overlapping or non-overlapping fields of view in an indoor office environment, with the layout shown in Figure 6.2. They are located over two floors, with four cameras on each floor, with a large 40 second gap between the floors. The areas not visible between cameras contain doors, corners and stairs ensuring no straight-line trajectories or linear velocities are possible between cameras. The eight time synchronised video feeds are fed into two P4 Windows PC in real-time. Figure 6.2 previously showed the layout the eight cameras over two floors.

6.8.2 Comparison to Batch learning processes

Most traditional methods of tracking objects inter camera, [30] [52], use a batch learning process to identify areas of interest, rather than incremental learning. A batch process technique is included within these results, to compare the effects on performance. After data collection, K-means is performed with five regions on each camera to cluster the entry/exit positions (see Figure 6.12). One disadvantage of the batch technique is that there is no accuracy improvement until all data is collected, while the use of an incremental learning algorithm allows the approach to increase accuracy over time as more data is collected. This also makes the proposed incremental technique more resistant to environmental variations as changes are incorporated over time. Also the K-means algorithm is dependant on upon k or the number of clusters which must be specified, this could cause multiple exits to be grouped incorrectly. The entry/exit regions resulting from the batch K-mean clustering technique can be seen in Figure 6.12(a), the larger the circle the more important the region. It can be seen that the incrementally learnt regions shown in Figure 6.12(b) are similar in position and importance (white is most important) to the batch learnt approach using all the data.

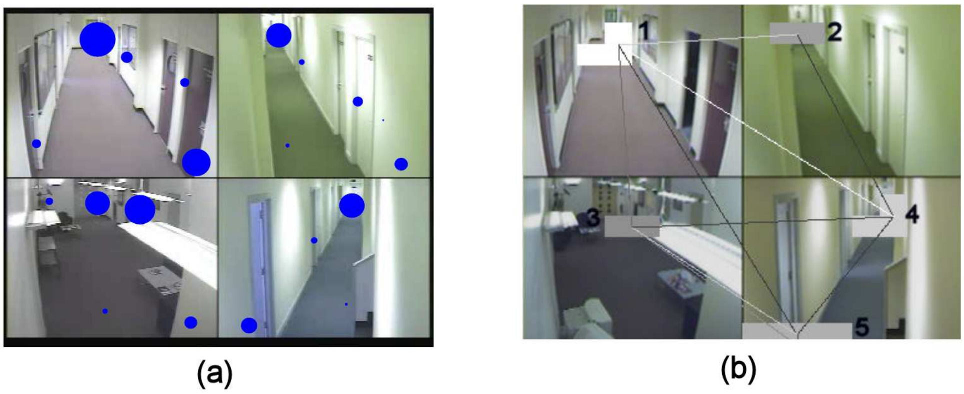


Figure 6.12: (a) shows the main entry and exit regions computed using a batch technique, size of the circles indicates the importance of entry/exit regions. While (b) shows the main entry and exit points using the incremental approach

6.8.3 4 Camera Setup

Initially the four cameras from the top floors were used for a detailed investigation into the learnt cues. There is no calibration of the camera environment with no *a priori* information about its environment. Over time additional information is incorporated. The experimental data was accumulated from 9am for 3 days

(72 hours), tracking a total of 7854 objects. Evaluation of the tracking was performed using two separate unseen ground-truthed 20 minute sequences each with 200 instances of people tracked for over one second. The two video sequences are quite different (see Figure 6.13 for examples of objects);

- **Test Video1**, This has a large number of new unique people, people walking in groups and the intra camera tracking failing intra camera due to erratic and slow movement.
- **Test Video2**, This consists of people moving cross camera, with fewer new unique people.

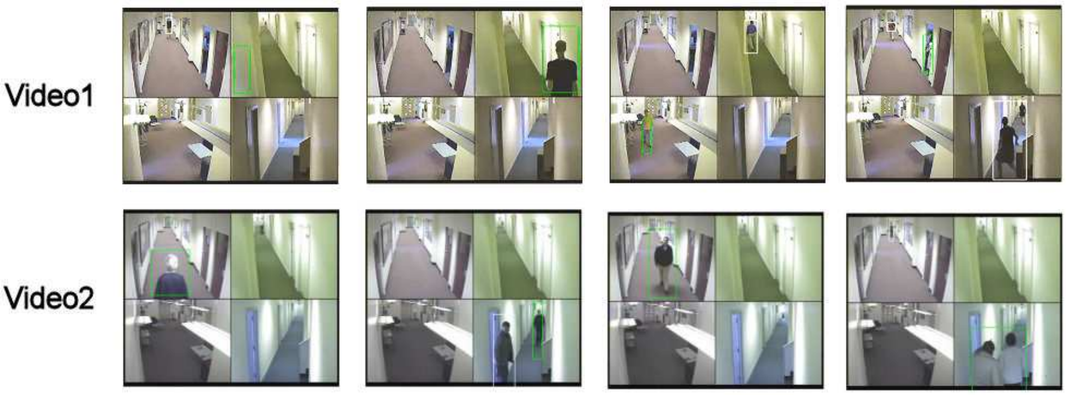


Figure 6.13: An example of some of the detected objects from both videos. The box indicates a tracked object.

Initially, the approach tracks using only the CLUT colour similarity between objects. Objects are tracked by maximising the posterior probability within a set time window, T. For these experiments, T is 40 seconds, where the cameras are close, provided the maximum peak exceeds the noise floor. While incrementally learning the inter camera relationships of the previously discussed weak cues, the system goes through a number of region size subdivisions. The first possible subdivision is after 1 hour, the next after 4 hours of operation, the third division is after 8 hours which corresponds to 1 full working day from 9am-5pm. The final possible subdivision level is reached after 32 hours which is two full working days. After 56 hours (three full working days) no further subdivisions take place but additional data is continually added to the prior until 72 hours (three full days) has passed. At each stage, the accuracy of all techniques for tracking; CLUT colour alone, posterior region links, posterior bounding box, and calibrated RGB colour, are measured. Table 6.2 shows all the single cues across the region subdivision sizes. The abbreviations for the similarity measures used are:

- HI(CLUT) - Histogram intersection of the CLUT colour descriptor.
- Reg - Maximising the posterior probability using the incrementally learnt prior on reappearance period.
- BB - Maximising the posterior probability using the learnt prior on object exit and entry size.
- HI(RGB) - Histogram intersection of the colour calibrated quantised RGB colour descriptor.
- Batch - Comparison technique of reappearance period prior computed using entry and exit regions derived through a batch processed K-means method.

Table 6.2 shows the initially poor performance of the individual descriptors. Over time accuracy improves, reaching 65% for Video2 using calibrated RGB after one working day or 8 hours. Note that after 8 hours, the priors are relatively stable and little benefit is gained from the addition of a further 2 days worth of observations.

At each stage of region refinement the accuracy of most techniques increases. After 72 hours and 7854 objects, each camera region has been subdivided at

Table 6.2: Table of results of using the individual descriptors with no fusion with subdivision of regions as additional data is accumulated with up to three days of data.

Video	Method	Accuracy:						
		Initial	Sub1	Sub2	Sub3	Sub4	Sub4	Sub4
	Time (Hr)	0	1	4	8	32	56	72
	Data (People)	0	367	1372	2694	5264	7612	7854
Video1	HI(CLUT)	50%	50%	50%	50%	50%	50%	50%
	Reg		33%	41%	45%	45%	44%	44%
	BB		42%	49%	55%	58%	60%	60%
	HI(RGB)	32%	45%	51%	53%	53%	55%	57%
Video2	HI(CLUT)	47%	47%	47%	47%	47%	47%	47%
	Reg		33%	40%	51%	51%	51%	52%
	BB		58%	64%	64%	64%	64%	64%
	HI(RGB)	40%	58%	62%	65%	65%	66%	67%

most 4 times, with Figure 6.14 showing the main entry/exit areas discovered by the approach.

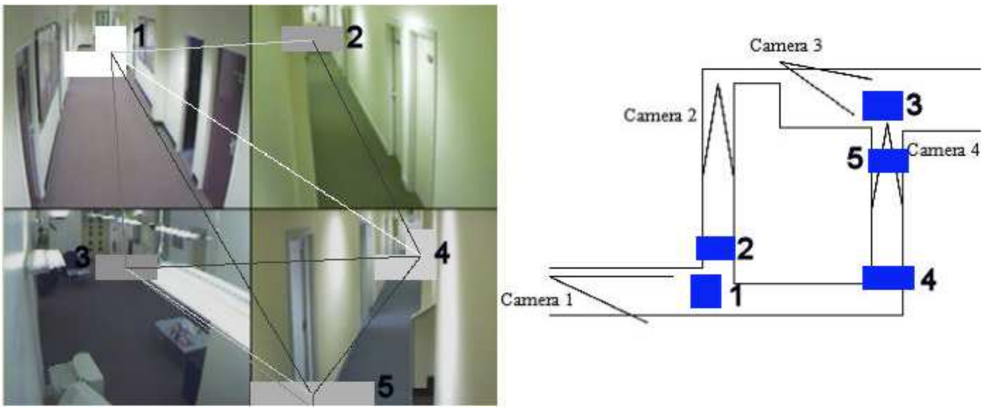


Figure 6.14: The main discovered entry and exit regions and a top down layout of the camera environment with these regions marked.

In order to increase the accuracy of inter camera tracking, we can fuse different descriptors together by multiplying the likelihoods as discussed in Section 6.4. This helps to remove some of the limitations of the individual cues. Table 6.3 shows the results of fusion over the same time period and subdivision intervals as Table 6.2. Some descriptors are not shown such as BB*HI(CLUT) as these performed worse than its colour calibrated equivalent BB*HI(RGB) which is shown.

Table 6.3 shows that the tracking accuracy has been increased from 50% to 73% and 47% to 79% on video1 and video2 respectively when using all three cues (BB*Reg*HI(RGB)) and 8 hours of data (1 working day). Combining all three weak cues together improves accuracy as it removes some of the limitations of each. Table 6.3 also shows that accumulating a further 2 days of data provides little improvement in accuracy over the cues constructed after only a day, demonstrating quick convergence upon a good solution. However, as data is accumulated, the posterior match becomes increasingly accurate and this can be used to provide a better correlation for the calculation of the priors. Table 6.4 uses one

Table 6.3: Table of results of using fusing the individual descriptors to increase tracking accuracy from start-up with up to a total of 3 days of data.

Video	Method	Accuracy:						
		Initial	Sub1	Sub2	Sub3	Sub4	Sub4	Sub4
	Time (Hr)	0	1	4	8	32	56	72
	Data (People)	0	367	1372	2694	5264	7612	7854
Video1	CLUT	50%	50%	50%	50%	50%	50%	50%
	Reg*CLUT		50%	55%	62%	61%	62%	64%
	Reg*RGB		55%	64%	68%	69%	69%	69%
	BB*Reg		49%	52%	55%	56%	56%	58%
	BB*RGB		59%	63%	67%	67%	67%	69%
	BB*Reg*RGB		57%	62%	71%	71%	73%	73%
	Batch	50%	50%	50%	50%	50%	50%	67%
Video2	CLUT	47%	47%	47%	47%	47%	47%	47%
	Reg*CLUT		60%	62%	72%	73%	74%	75%
	Reg*RGB		64%	66%	74%	75%	77%	77%
	BB*Reg		55%	57%	65%	66%	66%	66%
	BB*RGB		66%	72%	74%	76%	78%	78%
	BB*Reg*RGB		66%	72%	78%	79%	79%	79%
	Batch	47%	47%	47%	47%	47%	47%	76%

working day of data with three iterations. Each iteration results in an increase in accuracy allowing less false positive correlation to corrupt the three cues. However it can be seen that the benefits from this iterative refinement again stabilise quickly . The results of batch learnt links using k-means was combined with histogram intersection on CLUT (labelled Batch in the figure) and can therefore be directly compared with the results of Reg*HI(CLUT) in Table 6.3. Here it can be seen that batch learning only gives a marginal benefit over the incremental learning scheme. However, this slight increase in performance is only gained at subdivision 4 after 72 hours when all the data is available while the incremental scheme provides a gradual increase in performance as data is acquired.

Table 6.4, iteration 4 shows the results of the approach after the improvements of the earlier iterations with the extra data from all 3 days and provides only marginal improvements. These methods improve accuracy due to the minimisation of incorrect matches in the forming of region links and the other cues. This provides a final accuracy of 83% without data being added, or a small increase to 85% if two more days of data is used. Therefore using the extra 2 days data gives little or no improvement, again demonstrating the technique quickly converges on a stable solution after only 8 hours of camera monitoring. Figure 6.15 gives a visual representation of the accuracy increase over one day of data shown in Tables 6.2, 6.3 and further iterations in Table 6.4. The large increase in initial tracking accuracy from using only colour histogram intersection with the CLUT colour space can be seen. This large increase in accuracy fulfil the three ideals stated in the introduction, of working immediately, improving performance as additional data is captured, and an ability to adapt to environmental changes.

Table 6.4: Table of results of three iterations of the technique after one day of data, but with no new data collected. And results of using 3 days of data after iterating and refining accuracy.

Video	Method	Accuracy:			
		Iteration1	Iteration2	Iteration3	Iteration4
	Time (Hr)	8	8	8	72
	Data (People)	2694	2694	2694	7854
Video1	CLUT	50%	50%	50%	50%
	Reg*CLUT	64%	64%	64%	66%
	Reg*RGB	71%	72%	74%	73%
	BB*Reg	55%	57%	59%	58%
	BB*RGB	70%	70%	73%	73%
	BB*Reg*RGB	73%	70%	75%	77%
Video2	CLUT	47%	47%	47%	47%
	Reg*CLUT	72%	72%	72%	75%
	Reg*RGB	78%	79%	79%	77%
	BB*Reg	66%	65%	67%	66%
	BB*RGB	74%	77%	77%	78%
	BB*Reg*RGB	78%	83%	83%	85%

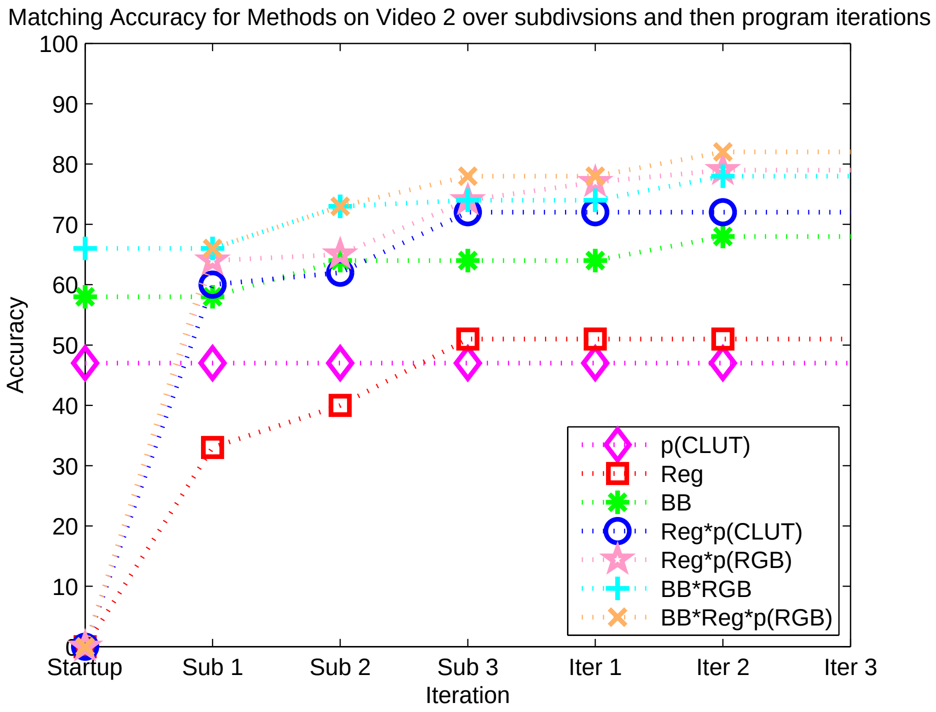


Figure 6.15: A graph showing the increasing accuracy with subsequent iterations of the methods using Video 2 up to iteration3.

6.8.4 Using the probably to rank the matches in order

Until now, the performance measure was based upon using the top ranked match against the correct person. As each possible correlation returns a likelihood score of a match, these can be presented in a ranked list. This reduces the quantity of data the operator has to process while improving accuracy. An example of this is in Figure 6.16, here the correct match is the third ranked. As all three results have a similar appearance with likelihoods, 0.15, 0.13 and 0.12 this indicates to the user that there is some uncertainty. This uncertainly is partly responsible for the ceiling accuracy of 85%. However, by considering the top three ranked correlations, the effective performance can be considered considerably higher.

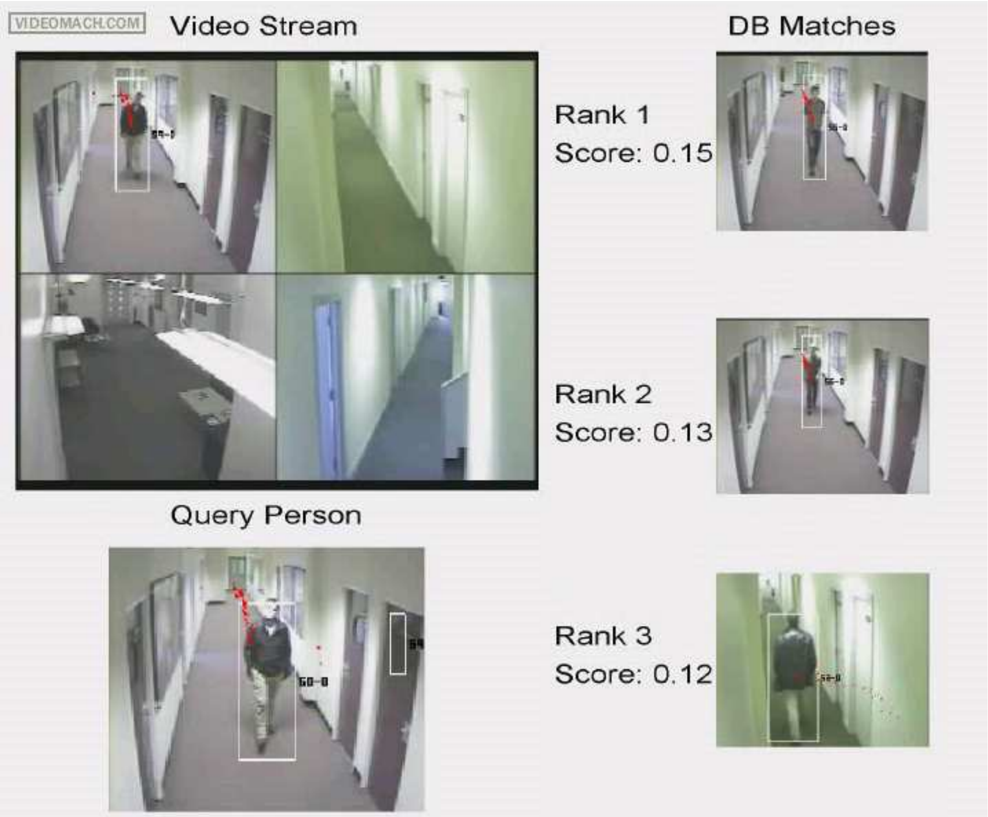


Figure 6.16: The rank of the best matches to the operator instead of the single optimum.

The graph in Figure 6.17, shows the result of this, based on the fusion of the bounding box, region links and histogram intersection of the RGB model. The incoming video stream is the top left image, with the lower left image showing the current query object. On the right are three ranked matches. Scoring using the top 3 ranked matches increases accuracy to over 90%.

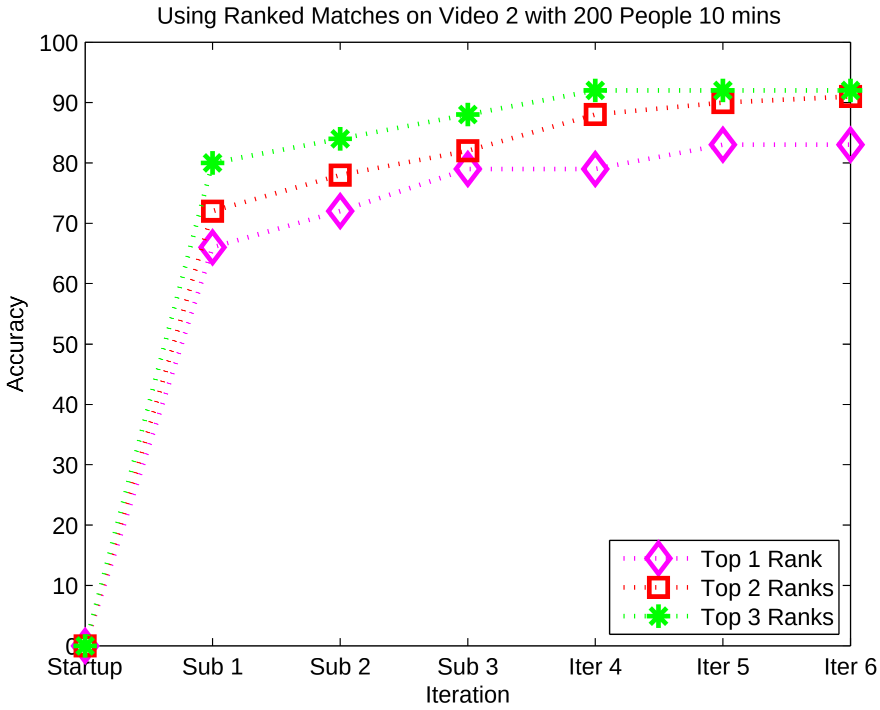


Figure 6.17: Using ranked matches to improve accuracy of tracking.

6.8.5 8 Camera Setup

To challenge and test the technique an eight camera setup is used. The eight camera setup introduces a large time gap between the two sets of four cameras. The larger time gap is to test if learning incremental relationships inter camera can operate with large temporal differences between the cameras. In addition within the large gap there are a number of different exits. This means a person leaving camera 3 is only around 30% likely to enter camera 8, while it is likely for a person leaving camera 4 to enter cameras 1 or 3. This could make it hard to form any distinct relationships between camera, affecting the overall tracking performance. However, the results show that this was not the case.

Due to the increased physical gap between the cameras on the two floors and that cameras 5,6, and 7 have overlapping fields of view, T, the reappearance period was adjusted to +/- 70seconds. This will caused a greater number of false positive entries to be added to the cues, however a longer period of data will be used to minimise this effect. To learn the relationships, a total of five days of data was used. A total of 40584 people were recorded who were tracked for more than 1 second. To test the performance, a one hour video sequence from the eight cameras was groundtruthed. This contained 400 people moving both intra and inter camera, in groups and singularly, Figure 6.18 shows a selection of frames from the test sequence.

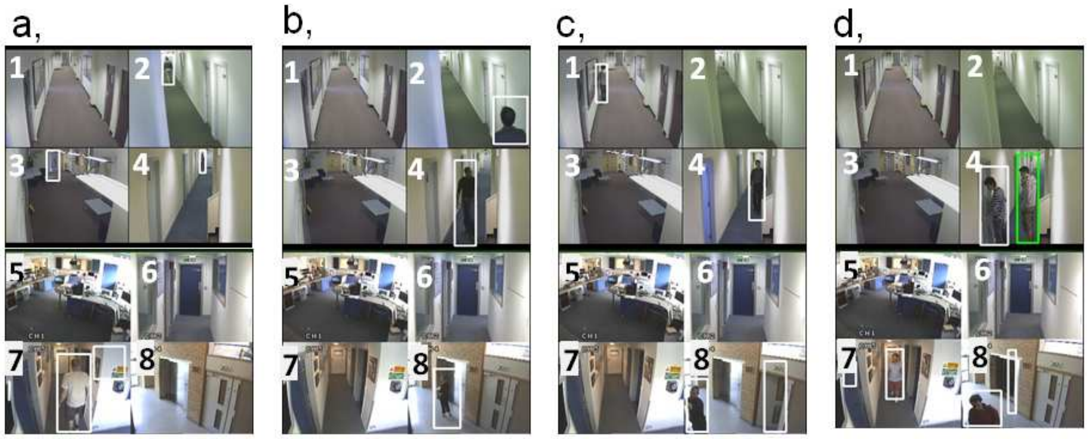


Figure 6.18: Example images from the test sequence.

Looking at camera 8 in Figure 6.18c and d there are false positive object tracks on the lift and door. These occurred on numerous occasion due to the limitations of the background segmentation. Figure 6.19 shows a subset of the resulting temporal relationship priors, between regions at a camera to camera level. Looking

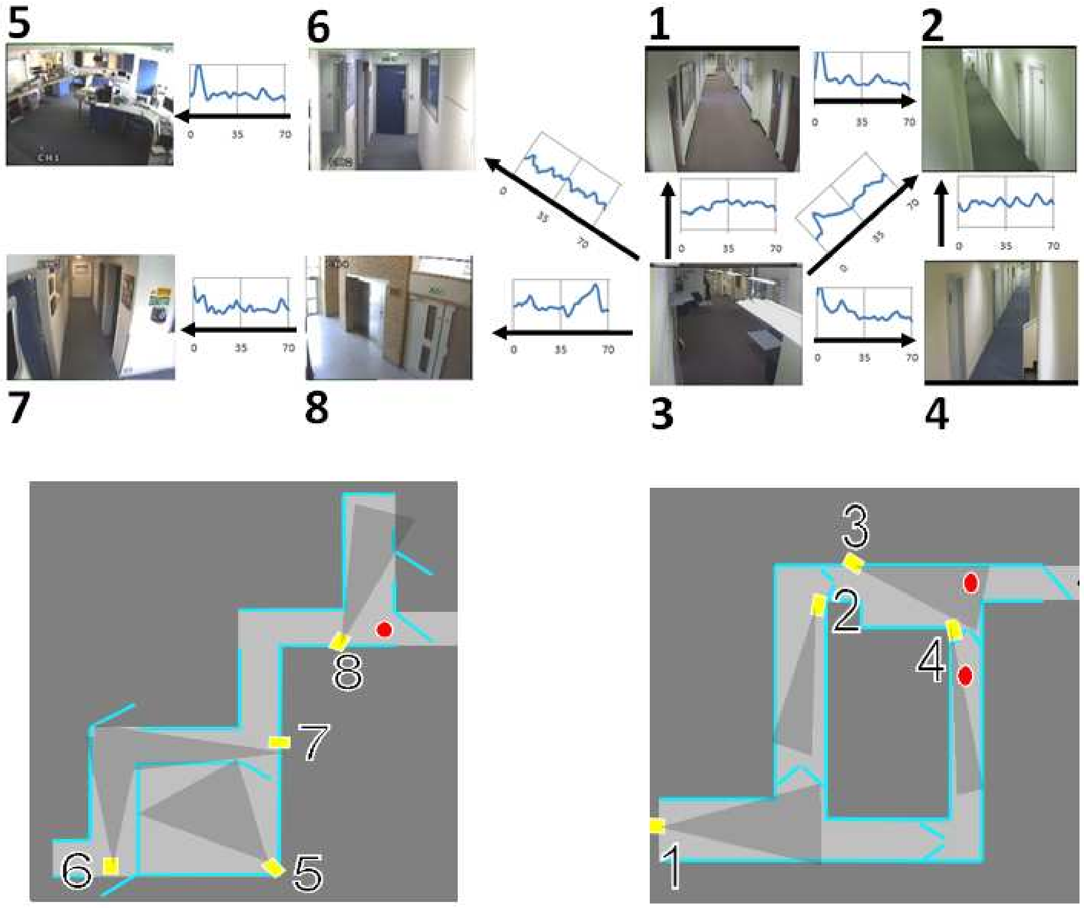


Figure 6.19: The temporal links between different cameras are shown, the X axis is time 0-70 seconds, the y axis the prior of $P(O_t|O_y)$ as shown in equation 6.7.

Table 6.5: Table of results of different cues using up to three iteration divisions of the regions with a total of 5 days footage. Test data is 400 people over 1 hour over all 8 cameras.

Video	Method	Accuracy:						
		Initial	Sub1	Sub2	Sub3	Sub4	Sub4	Sub4
	Time (Hr)	0	1	4	8	48	96	120
	Data (People)	0	367	1372	8550	15250	31241	40584
Video3	CLUT	39%	39%	39%	39%	39%	39%	39%
	Reg		53%	60%	66%	66%	66%	66%
	BB		49%	54%	56%	59%	59%	59%
	RGB	33%	47%	55%	58%	59%	59%	59%
	BB*Reg*RGB		65%	68%	74%	76%	76%	76%
	Batch							69%

at Figure 6.19 the peaks shown in the priors such as between cameras 3 to 8 and 3 to 2 show there is a strong link between these cameras. While the flatter links between cameras 3 to 6 or 4 to 2, show there is no direct relationship between the cameras. Table 6.5 shows the results of the groundtruth data on the 5 days of footage. Performance is shown at various stages over the five days, with up to 3 possible region sub division occurring if a strong link has been found between regions.

The results of the 8 camera tracker shown in table 6.5 are also displayed in graph form in Figure 6.20. Looking at the results, the overall performance of the tracker increases from 30% to 76%, and this occurs within two days of data, with the performance stable for the remaining three days. It is interesting to note that the temporal region linking cue (Reg) alone has a performance of 66% after 8 hours of

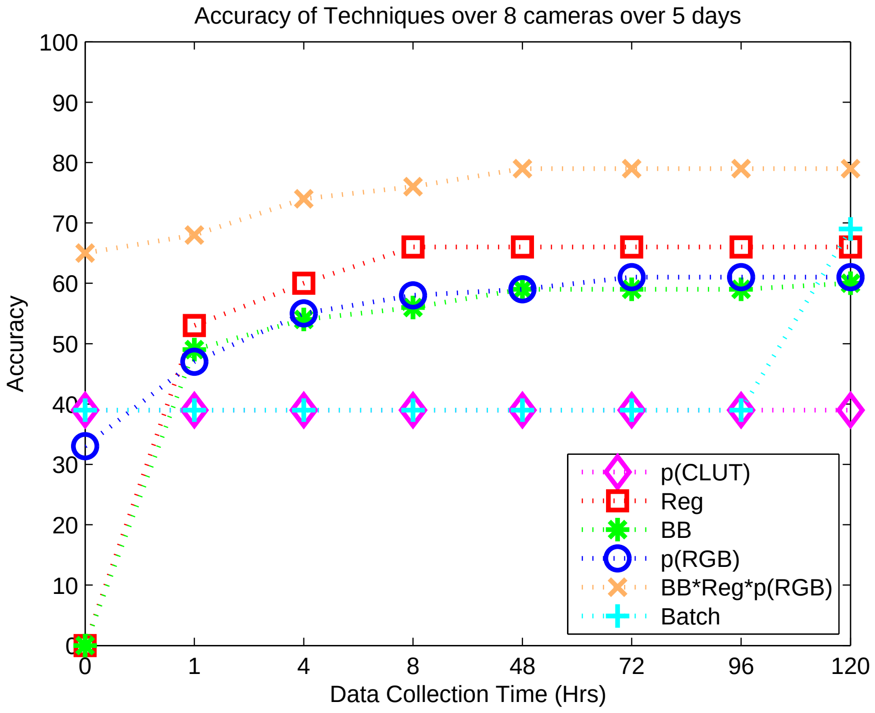


Figure 6.20: Accuracy of inter camera tracker using up to 5 days data.

Table 6.6: Table of results with the top 3 matches are used to match with the tracked person. All the different cues using up to three iteration divisions of the regions with a total of 5 days footage.

Video	Method	Accuracy:						
		Initial	Sub1	Sub2	Sub3	Sub4	Sub4	Sub4
	Time (Hr)	0	1	4	8	48	96	120
	Data (People)	0	367	1372	8550	15250	31241	40584
Video3	CLUT	44%	44%	44%	44%	44%	44%	44%
	Reg		68%	74%	79%	80%	81%	81%
	BB		57%	59%	64%	65%	65%	65%
	RGB	45%	50%	60%	61%	64%	64%	64%
	BB*Reg*RGB		75%	81%	89%	91%	91%	91%
	Batch							82%

data. This has no size or appearance information, only using the temporal prior to compute where the object is likely to have come from. The success of this over appearance alone shows how important it is to learn patterns of activity, especially when false positive detections (as in Figure 6.18c and d, camera 8) would cause the appearance correlation to fail.

To improve performance, the tracking was extended in a similar way to that in Section 6.8.4, to use the top three possible matches instead of just the first. Table 6.6 and graph 6.21 show the results, over the same 5 days of footage, where instead a success is if the person is correctly found in the top three matches.

Using the ranked top three matches increases the performance from a low 44% to a high of 91% using all three cues of colour, temporal and size to weight a simple appearance correlation. A batch technique is unable to increased performance

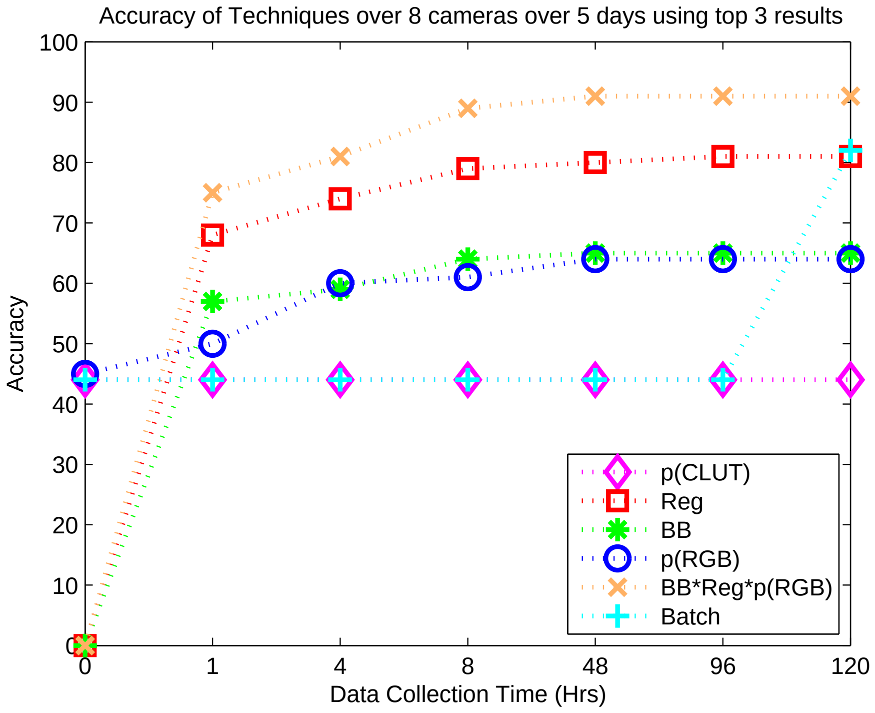


Figure 6.21: Accuracy of inter camera tracker using up to 5 days data.

until all data has been collected, and then achieves 82% accuracy. Looking at the tracking accuracy of individual cues in Table 6.5, the temporal relationships between regions on the cameras is influential, with it alone being able to label 81% of people inter and intra camera correctly. This is important as the colour matching is heavy affected by the moving doors and lighting changes that occurs on cameras 5 to 8.

6.9 Conclusion

This chapter has shown a technique that is able to learn relationships between regions within cameras to track people between up to eight cameras over a wide area indoor site. The use of three individually simple and weak cues allows for the technique to be run in real time (25fps) and when fused together a powerful prior for object tracking is created. This works with no *a priori* information or calibration of the cameras. Examining the ideals of a real time tracker presented in the introduction of the chapter. If can be seen that this method addresses all the statements. Upon initialisation it is able to work immediately, its performance dramatically improves as new evidence becomes available, and due its subdivision and re-examining of the regions is adaptable to changes in the camera's environment. The method has been tested on three different manually groundtruthed video sequences using up to eight camera with overlapping and non-overlapping fields of view. The technique has also demonstrated that the cross camera region linking can operate with cameras of considerable separation, where objects may take up to 40 seconds to reappear on another camera.

Chapter 7

Discussion and future work

The aim of this work was to track and maintain the identity of moving objects on cameras in a scalable approach, that adapts to the camera environment. Solutions for a number of challenging problems within the field of object tracking have been presented. A technique to track and maintain an object's identity through a scene of crowded people was presented, together with a number of solutions to allow people to be tracked between multiple, uncalibrated cameras. Both techniques were based on an object's appearance descriptor, therefore, an in depth investigation into possible methods of constructing appearance descriptors to ensure optimum efficiency and detail was conducted.

Three different methods used in the construction of appearance descriptors were examined in chapter 4. The size of the quantisation, the colour space and the correlation methods, were examined for both intra and inter camera tracking contexts. In both intra and inter camera environments, the Bhattacharyya coefficient measure provided good correlation between the true positive results, while having a good separation from the false positive results. The use of Parzen-Windowing when constructing RGB histograms reduces both over and under fitting of data due to incorrect bin size selected for the number of samples available. With the use of an In-Parzen window during construction, RGB gives the highest performance of the colour spaces, improving overall performance of correlation across the complete range of bin sizes. For an application with real-time constraints a compromise using the manually defined Colour Lookup Table colour space was demonstrated to have a similar performance. However, at times, the low quantisation size, means the histograms are underpopulated, causing intermittent correlation failures.

A technique to track individuals within a crowded scene was presented in chapter 5. Novelly, it used the strengths of separate global and local methods fused with dynamic programming. A head and shoulder detector with a global search area was used to detect the location of people. The responses of the head and shoulder detector are taken as observations of the human hypotheses. These were tracked with a localised frame-by-frame Mean Shift optimisation until the appearance of the tracked object is deemed corrupted due to occlusion or drift. These short tracklets produce an over complete trajectory path of all people. Dynamic programming is used to find the least-cost path through the sequence. Two additional learnt models of human motion refine the least-cost path. These apply motion constraints at a pixel level and detection level to identify and remove outliers. Four different and challenging sequences of people interacting and occluding were used to test the technique. Promising results for many of the people were shown for the single uncalibrated camera sequences.

In addition to the tracking intra camera, work to bridge the gaps between unconnected non-overlapping uncalibrated cameras was presented in chapter 6. A technique to learn how people's appearance and movement relate between cameras was used to improve basic observation likelihood correlation. Up to 8 cameras were used with simple weak descriptor cues to enable real-time operation (25fps). The weak cues were fused together with the objects appearance to form a powerful correlation descriptor. No pre-calibration or environmental information was provided to the cameras, making the technique ideal for larger networks of cameras. The incremental learning allows for an improvement in performance throughout learning. This is unlike a batch process, as a batch process can only work once all data has been collected. The constant refinement of the camera relationship cues, allows for continuous performance improvement during operation.

There is no *a priori* information or calibration of the cameras. Upon initialisation it is able to work immediately, its performance dramatically improves as new evidence becomes available, and due to its subdivision and re-examining of the regions is adaptable to changes in the camera's environment. The method has been tested on three different videos, these are manually groundtruthed sequences using up to eight camera with both overlapping and non-overlapping fields of view.

Generic techniques with novel contributions to track and correlate moving objects on two different challenging environments have been presented. The intra camera tracking of people within complex crowd interactions is possible with the two part approach of detection and tracking. While an incrementally learnt approach of inter camera tracking over eight cameras separated up to 40 seconds in real time with no calibration or a *priori* information has been shown to correlate object successfully.

7.1 Future Work

In order to take the theorems and techniques represented within this work further, it could be applied to an outdoor environment, which has scope to be much larger. After the successful test on linking camera between two floors (40 seconds apart), it should be possible to link further cameras. Future work would investigate the limitations of trying to extend the techniques to larger external deployments over square miles of urban environment where both people and vehicles operate and interact. Most of the problems this would create would be within the low-level intra camera tracking. Including problems, such as increased shadows, and increased numbers of occluded people, while the pixel resolution of the overall moving objects would be reduced. This would reduce the samples in the appearance histograms, however the Parzen windowing or CLUT methods are designed to compensate for this.

The actual higher level learnt cues described in the inter camera tracking work, are generic in their design allowing them to be applied to this outdoor work with little modification. In addition, it would be interesting to test the cues within a different object field or environment. For example learning the relationships between road surveillance cameras, similar to the work of Huang and Russell [42]. The traffic patterns could be learnt for *normal* traffic, allowing for abnormalities to the camera relationship model to be identified and flagged. An implementation of the technique within an underground station would allow monitoring of a wide area. However for this to be effective, accurate multiple person tracking would be required.

Within the work for the crowd tracking despite the encouraging results, there is much scope for future improvement. The main reason for failures was the lack of tracklets covering trajectories, - this was due to two problems. Firstly, if a person was occluded for a long period behind someone else, their best path would fail to correlate again once the occlusion ended as the motion and colour models would fail. Secondly despite the high detection rate of the head and shoulder detector, it produces many false negative results for people facing away from the camera, and without these seed locations, no tracklets can be produced. To solve this, part body detectors could be implemented similar to that proposed by Wu and Nevatia [111] to provide additional seed positions. To further enhance tracking, the motion of features within the crowds could be used, Browstow and Cipolla [19] and Rabaud and Belongie [87], both use clustering of local features to count people within a crowd. In the current system, the human head and shoulder detector, is learnt off-line, using manually found training examples. However during the learning of the human motion motions, general detectors could be trained on specific examples found in the training period. This could achieve both higher accuracy as detection would be tuned to the environment. Another method to increase speed, would be to relate the tracking back to the detection, as the tracking optimisation could restrain the detector to the surrounding neighbourhood.

Bibliography

[1] CAVIAR: Context Aware Vision using Image-based Active Recognition, EC project/ist 2001 37540. *http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.*

[2] "Mathworks". *http://www.mathworks.com/access/helpdesk/help/toolbox/images/hsvcone.*

[3] PETS: Performance Evaluation of Tracking and Surveillance. *http://www.cvg.cs.rdg.ac.uk/slides/pets.html.*

[4] Mittal A. and L. Davis. *In International Journal of Computer Vision*, 51(3):189–203, 2003.

[5] F. Aherne, N. Thacker, and P. Rockett. "The Bhattacharyya Metric as an Absolute Similarity Measure for Frequency Coded Data". *In Kybernetika*, 32(4):1–07, 1997.

[6] H. Andrews. *"Introduction to Mathematical Techniques in Pattern Recognition"*. R.E. Krieger Pub. Co, 1983.

[7] J. Annesley and J. Orwell. "On the Use of MPEG-7 for Visual Surveillance". *In Proc. of 6th IEEE International Workshop on Visual Surveillance*, 2006.

[8] O. Arandjelovic and A. Zisserman. "Automatic Face Recognition for Film Character Retrieval in Feature-Length Films". *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1:860–867, 2005.

[9] R. Bellman. "On a Rounting Problem". *Quart. Appl. Math*, pages 87–90, 1985.

[10] B. Berlin and P. Kay. "Basic Color Terms : Their Universality and Evolution.". *Paperback ed. Berkeley ; Oxford: University of California*, pages 196–201, 1991.

[11] Q. Beymer. "Person Counting using Stereo". *In Proc. of Workshop on Human Motion*, pages 127–133, 2000.

[12] J.K. Black, T.J. Ellis, and D. Makris. "Wide Area Surveillance with a Multi-Camera Network". *In Proc. of Intelligent Distributed Surveillance Systems (IDSS-04)*, pages 21–25, 2003.

[13] J.K. Black, T.J Ellis, and P. Rosin. "Multi-View Image Surveillance and Tracking". *In Proc. of IEEE Workshop on Motion and Video Computing*, pages 169–174, 2002.

[14] B. Bose, X Wang, and E. Grimson. "Detecting and Tracking Multiple Interacting Objects Without Class-Specifc Models". *In Technical report MIT-CSAIL-TR-2006-027 Massachusetts Institute of Technology*, 2006.

[15] R. Bowden, A. Gilbert, and P. KaewTraKulPong. "Tracking Objects Across Uncalibrated Arbitrary Topology Camera Networks, ". *Intelligent Distributed Video Surveillance, Systems S.A Velastin and P Remagnino (Eds)*, pages 157–183, 2005.

[16] R. Bowden and P. KaewTrakulPong. "Towards Automated Wide Area Visual Surveillance: Tracking Objects Between Spatially Separated, Uncalibrated Views". *In Proc. Vision, Image and Signal Processing*, 152(2):213–224, 2005.

[17] Y. Boykov and D. Huttenlocher. “Adaptive Bayesian Recognition in Tracking Rigid Objects”. *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR’00)*, pages 697–704, 2000.

[18] G.R. Bradski. “Computer Vision Face Tracking as a Component of Perceptual User Interface”. *In Proc. of Workshop on Applications of Computer Vision*, pages 214–219, 1998.

[19] G. Browstow and R. Cipolla. “Unsupervised Bayesian Detection of Independent Motion in Crowds”. *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR’05)*, pages 594 – 601, 2005.

[20] Q. Cai and J. Agrarian. “Tracking Human Motion using Multiple Cameras”. *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR’96)*, pages 67–72, 1996.

[21] T. Chang and S. Gong. “Bayesian Modality Fusion for Tracking Multiple People with a Multi-Camera System”. *In Proc. of European Workshop on Advanced Video-based Surveillance Systems*, 2001.

[22] T.H. Chang, S. Gong, and E. Ong. “Tracking Multiple People under Occlusion using Multiple Cameras”. *In Proc. of BMVA British Machine Vision Conference (BMVC’00)*, pages 566–575, 2000.

[23] A. Chilgunde, P. Kumar, S. Ranganath, and H. WeiMin. “Multi-Camera Target Tracking in Blind Regions of Cameras with Non-overlapping Fields of View”. *In Proc. of BMVA British Machine Vision Conference (BMVC’04)*, pages 1–10.

[24] D. Comaniciu, V Ramesh, and P. Meer. “Kernel-Based Object Tracking”. *In IEEE Transactions Pattern Analysis and Machine Intelligence*, 25:564–577, 2003.

[25] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. “Detecting Objects, Shadows and Ghosts in Video Streams by Exploiting Color and Motion Information”. *In Proc. of 11th International Conference on Image Analysis and Processing (CIAP'01)*, pages 360–369.

[26] A. Dick and M. Brooks. “A Stochastic Approach to Tracking Objects Across Multiple Cameras”. *In Proc. of Australian Conference on Artificial Intelligence*, pages 160–170, 2004.

[27] E.W. Dijkstra. “A note on Two Problems in Connexion with Graphs ”. *Numerische Mathematik*, 1, 1959.

[28] S.L. Dockstader and A.M. Tekalp. “Multiple Camera Tracking of Interacting and Occluded Human Motion”. *In Proc. of IEEE*, 89(10):1441–1455, 2001.

[29] N. Dowson and R. Bowden. “A Unifying Framework for Mutual Information Methods for use in Non-Linear Optimisation.”. *In Proc. of European Conference on Computer Vision (ECCV'06)*, I:365–378, 2006.

[30] T.J. Ellis, D. Makris, and J.K. Black. “Learning a Multi-Camera Topology”. *In Proc. of Joint IEEE Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, pages 165–171, 2003.

[31] M. Felsberg and G. Granlund. “P-Channels: Robust Multivariate M-Estimation of Large Datasets”. *In Proc. of International Conference on Pattern Recognition (ICPR'06)*, III:262–267, 2006.

[32] P. Figueroa, N. Leite, R. Barros, I. Cohen, and Medioni G. “Tracking Soccer Players using the Graph Representation”. *In Proc. of International Conference on Pattern Recognition (ICPR'04)*, pages 787–790.

[33] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. "Multi-Camera People Tracking with a Probabilistic Occupancy Map". *to be published In IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.

[34] D. Forsyth and M. Fleck. "Body Plans". *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR97)*, pages 678–683.

[35] J. Giebel, D Gavrilla, and C. Schnorr. "A Bayesian Framework for Multi-cue 3d Object Tracking". *In Proc. of European Conference on Computer Vision (ECCV'04)*, IV:241–252, 2004.

[36] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. "Novel Approach to Nonlinear/Non-Gaussian Bayesian Sate Estimation". *In Proc. of IEE Radar and Signal Processing*, pages 107–113, 1993.

[37] D. Gorodnichy. "On Importance of Nose for Face Tracking ". *In Proc. of IEEE International Conference on Automatic Face and Gesture Recognition (FG'02)*, pages 188–196, 2002.

[38] U. Grenander, Y. Chow, and D.M. Keenan. "HANDS. A Pattern Theoretical Study of Biological Shapes". *In Springer-Verlag, Berlin, Heidelberg*, 1991.

[39] E. Hadjidemetriou, M.D. Grossberg, and S.K. Nayar. "Multi Resolution Histograms and their use for Recognition". *In Transactions on Pattern Analysis and Machine Intelligence*, 26(7):831–847, 2004.

[40] I. Haritaoglu, D. Harwood, and L.S. Davis. "W4: Who? When? Where? What? a Real-Time System for Detecting and Tracking People". *In Proc. of IEEE International Conference on Automatic Face and Gesture Recognition (FG'98)*, pages 222–231, 1998.

[41] T. Horprasert, D. Harwood, and L.S. Davis. "A Statistical Approach for Real-Time Robust Background Subtraction and Shadow Detection". *In Proc. Frame-Rate Applications Workshop*, pages 119–127, 1999.

[42] T. Huang and S. Russell. "Object Identification in a Bayesian Context". *In Proc. of International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 1276–1283, 1997.

[43] C. Hue, J.L. Cadre, and P. Perez. "Tracking Multiple Objects with Particle Filtering". *In Trans on Aerospace and Electronic Systems*, 38:313–318, 2003.

[44] A. Ilie and G. Welch. "Ensuring Color Consistency across Multiple Cameras". *Techincal Report TR05-011*, 2005.

[45] S.S. Intille, J.W. Davis, and A.F. Bobick. "Real-Time Closed-World Tracking". *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pages 697–703, 1997.

[46] S. Ioffe and D. Forsyth. "Probabilistic Methods for Finding People". *In International Journal of Computer Vision*, 43(1):45–68, 2001.

[47] M. Isard and A. Blake. "CONDENSATION: Conditional Density Propagation for Visual Tracking". *In Internatinal Journal on Computer Vision*, 29(1):5–28, 1998.

[48] M. Isard and J. MacCormick. "BramMBLe: A Bayesian Multiple Blob Tracker". *In Proc. of IEEE International Conference on Computer Vision and Pattern Reconition (CVPR'01)*, 2:34–41, 2001.

[49] Y. Ivanov, C. Stauffer, A. Bobick, and W.E.L. Grimson. "Video Surveillance of Interactions". *In Proc. of CVPR'99 Workshop on Visual Surveillance*, 1999.

[50] S. Iwase and H. Saito. "Parallel Tracking of all Soccer Players by Intergrating Detected Postions in Multiple View Images.". *In Proc. of International Conference on Pattern Recognition (ICPR'04)*, pages 751–754.

[51] O. Javed, Z. Rasheed, K. Shafique, and M. Shah. "Tracking Across Multiple Cameras with Disjoint Views". *In Proc. of IEEE International Conference on Computer Vision (ICCV'03)*, pages 952–957, 2003.

[52] P. KaewTrakulPong and R. Bowden. "A Real-time Adaptive Visual Surveillance System for Tracking Low Resolution Colour Targets in Dynamically Changing Scenes". *In Journal of Image and Vision Computing*, 21(10):913–929, 2003.

[53] T. Kailath. "The Divergence and Bhattacharyya Distance Measures in Signal Selection". *In IEEE Transactions on Communication Technology*, 15(1):52–60.

[54] J. Kang, I. Cohen, and G. Medioni. "Tracking People in Crowded Scenes across Multiple Cameras". *In Proc. of Asian Conference on Computer Vision (ACCV'04)*, 2004.

[55] J.K. Kearney, W.B. Thompson, and D.L. Boley. "Optical Flow Extimation: An Error Analysis of Gradient based Methods with Local Optimization". *In Transactions on Pattern Analysis and Machine Intelligence*, 9:229–244, 1987.

[56] P. Kelly, A. Katkere, D. Kuramura, S. Moezzi, and S. Chatterjee. "An Architecture for Multiple Perspective Interactive Video". *In Proc. of the 3rd ACE International Conference on Multimedia*, pages 201–212, 1995.

[57] V. Kettnaker and R. Zabih. "Bayesian Multi-Camera Surveillance". *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'99)*, pages 253–259, 1999.

[58] S. Khan and M. Shah. "Tracking People in Presence of Occlusion". *In Proc. of Asian Conference on Computer Vision, (ACCV'00)*, 2000.

[59] Z. Khan, T. Balch, and F. Dellaert. "An MCMC-Based Particle Filter for Tracking Multiple Interacting Targets". *In Proc. of European Conference on Computer Vision (ECCV'04)*, IV:279–290, 2004.

[60] G. Kitagawa. "Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models". *In Journal of Computational and Graphical Statistics*, 5(1), 1996.

[61] J.J. Koenderink and A.J. Van-Doorn. "Blur and Disorder". *In Journal of Visual Communication and Image Representation*, 11(2):237–244, 2000.

[62] D. Koller, J. Weber, and J. Malik. "Robust Multiple Car Tracking with Occlusion Reasoning". *In Proc. of European Conference on Computer Vision (ECCV'94)*, pages 186–196, 1994.

[63] B. Lucas and T Kanade. "An Iterative Image Registration Technique with an Application to Stereo Vision". *In Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1998.

[64] S. Mallat. "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation.". *In Transactions on Pattern Analysis and Machine Intelligence*, 11(2):674–693, 1989.

[65] S. McKenna, S. Jabri, Z. Duric, and H. Wechsler. "Tracking Interacting People". *In Proc. of IEEE International Conference on Automatic Face and Gesture Recognition (FG'00)*, pages 348–353, 2000.

[66] A. Micilotta, E. Ong, and R. Bowden. "Detection and Tracking of Humans by Probabilistic Body Part Assembly". *In Proc. of BMVA British Machine Vision Conference (BMVC'05)*, I:429–438, 2005.

[67] I. Mikic, S. Santini, and R. Jain. “Video processing and Integration from Multiple Cameras”. *In Proc. of Image Understanding Workshop*, 1998.

[68] K. Mikolajczyk, B. Leibe, and B. Schiele. “Multiple Object Class Detection with a Generative Model”. *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR’06)*, I:26–36, 2006.

[69] K. Mikolajczyk, C. Schmid, and A. Zisserman. “Human Detection Based on a Probabilistic Assembly of Robust Part Detector”. *In Proc. of European Conference on Computer Vision (ECCV’04)*, I:69–82, 2004.

[70] T. Misu, M. Naemura, Z. Wentao, Y. Izumi, and K. Fukui. “Robust Tracking of Soccer Players Based on Data Fusion”. *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR’02)*, pages 556–561, 2002.

[71] C. Mohan, A. Papageorgiou and T. Poggio. “Example-based Object Detection in Images by Components”. *In Transactions on Pattern Analysis and Machine Intelligence*, 23(4):349–361, 2001.

[72] V.I Morariu and O.I Camps. “Modeling Correspondences for Multi-Camera Tracking using Nonlinear Manifold Learning and Target Dynamics”. *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR’06)*, I:545–552, 2006.

[73] Joshi N. “Color Calibrator for Arrays of Inexpensive Image Sensors”. *MS Thesis, Stanford University Department of Computer Science*, 2004.

[74] C.J. Needham and R.D. Boyle. “Tracking Multiple Sports Players through Occlusion, Congestion and Scale”. *In Proc. of BMVA British Machine Vision Conference (BMVC’01)*, I:93–102, 2001.

[75] P. Nillius, J. Sullivan, and S. Carlsson. “Multi-Target Tracking - Linking Identities using Bayesian Network Inference”. *In Proc. of IEEE*

International Conference on Computer Vision and Pattern Recognition (CVPR'98), pages 2187 – 2194, 2006.

[76] K. Nummiaro, E. Koller-Meier, T. Svoboda, D. Roth, and L. VanGool. "color-based object tracking in multi-camera environments". *In Proc. of Deutsche Arbeitsgemeinschaft fur Mustererkennung (DAGM'03)*, pages 591 – 599, 2003.

[77] K. Nummiaro, E. Koller-Meier, and L. VanGool. "An Adaptive Color-Based Particle Filter". *In Journal of Image and Vision Computing*, 21(1):99 – 110, 2003.

[78] K. Okuma, A Taleghani, N. DeFreitas, J.J. Little, and D.G. Lowe. "A Boosted Particle Filter: Multitarget Detection and Tracking". *In Proc. of European Conference on Computer Vision (ECCV'04)*, pages 28–39, 2004.

[79] M. Oren, C.P. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. "Pedestrian Detection using Wavelet Templates.". *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition(CVPR'97)*, pages 193–199, 1997.

[80] J. Orwell, S. Massey, P. Remagnino, D. Greenhill, , and G.A. Jones. "A Multi-Agent Framework for Visual Surveillance". *In Proc. of International Conference on Image Analysis and Processing (IAPR)*, pages 1104–1107, 1999.

[81] J. Orwell, P. Remagnino, and G.A. Jones. "Multi-Camera Color Tracking". *In Proc. of 2nd IEEE Workshop on Visual Surveillance*, pages 1104–1107, 1999.

[82] C.P. Papageorgiou, M. Oren, and T. Poggio. "A General Framework for Object Detection". *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'98)*, pages 552–562, 1998.

[83] E. Parzen. “On Estimation of a Probability Density Function and Mode”. *In Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[84] H. Pasula, S. Russell, M. Ostland, and Y. Ritov. “Tracking many Objects with many Sensors”. *In Proc. of International Joint Conferences on Artificial Intelligence*, pages 1160–1171, 1999.

[85] P. Perez, C. Hue, J. Vermaak, and M. Gannet. “Color-Based Probabilistic Tracking”. *In Proc. of European Conference on Computer Vision (ECCV'02)*, pages 661–675, 2002.

[86] F. Porikli. “Inter-Camera Color Calibration by Cross-Correlation Model Function”. *IEEE International Conference on Image Processing (ICIP)*, II:133–136, 2003.

[87] V. Rabaud and S. Belongie. “Counting Crowded Moving Objects”. *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR06)*, I:705–711, 2006.

[88] C. Ridder, O. Munkelt, and H. Kirchner. “Adaptive Background Estimation and Foreground Detection using Kalman Fltering”. *In Proc. of International Conference on recent Advances in Mechatronics (ICAM)*, pages 193–199, 1995.

[89] P. Rosin and T. Ellis. “Image Difference Threshold Strategies and Shadow Detection”. *In Proc. of BMVA British Machine Vision Conference (BMVC'95)*, pages 347–356, 1995.

[90] H.A. Rowley, S. Baluja, and T. Kanade. “Neural Network-based Face Detection.”. *In Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[91] R. Schapire, Y. Freund, P. Bartlett, and W. Lee. “Boosting the Margin: A new Explanation for the Effectiveness of Voting Methods.”. *In The Annals of Statistics*, 26(5):1651–1686, 1998.

[92] A. Senior. “tracking people with probabilistic appearance models”. *In Proc. of IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 48–55, 2002.

[93] C.E. Shannon. “A Mathematical Theory of Communication.”. *In Bell System Technical Journal*, 27:379–423, 1948.

[94] V. Shet, D. Harwood, and L. Davis. “Multivalued Default Logic for Identity Maintenance in Visual Surveillance”. *In Proc. of European Conference on Computer Vision (ECCV’06)*, pages 119–132.

[95] L. Sigal, B.H. Isard, M. Sigelman, and M.J. Black. “Attractive People: Assembling Loose-Limbed Models using non-Parametric Belief Propagation”. *In Proc. of Neural Information Processing System Conference (NIPS)*, 2003.

[96] E.P. Simoncelli, E.H. Adelson, and D.J. Heeger. “Probability Distributions of Optical Flow”. *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR’91)*, pages 310–315, 1991.

[97] A. Smith and J. Blinn. “Blue Screen Matting”. *In Proc. of Conference of Computer graphics and interactive techniques (SIGGRAPH ’96)*, pages 259–268, 1996.

[98] GretagMacbeth Color Management Solutions. *www.gretagmacbeth.com.*

[99] C. Stauffer. “Estimating Tracking Sources and Sinks”. *In Proc. of Event Mining Workshop*, 2003.

[100] C. Stauffer and W.E.L. Grimson. “Learning Patterns of Activity using Real-time Tracking”. *In Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.

[101] C. Studholme, D.L.G. Hill, and D.J. Hawkes. “automated 3d registration of mr and ct images of the head”. *In Medical Image Analysis*, (2).

[102] J. Sturges and T.W.A. Whitfield. “Locating Basic Colours in the Munsell Space”. *In Color Research and Application*, 20(6):364–376, 1995.

[103] M.J. Swain and D.H. Ballard. “Color Indexing”. *In International Journal of Computer Vision*, 7(1), 1991.

[104] M.M. Trivedi, I. Mikic, and S.K. Bhonsle. “Active Camera Networks and Semantic Event Databases for Intelligent Environments”. *In Proc. IEEE Workshop on Human Modelling, Analysis and Synthesis*, 2000.

[105] V. Vapnik. “The Nature of Statistical Learning Theory”. *Springer Verlag*, 1995.

[106] J. Vermaak, A. Doucet, and P. Perez. “Maintaining Multi-Modality through Mixture Tracking.”. *In Proc. of IEEE International Conference on Computer Vision (ICCV’03)*, pages 1110–1116, 2003.

[107] P. Viola and M. Jones. “Rapid Object Detection using a Boosted Cascade of Simple Features”. *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR’01)*, I:511–518, 2001.

[108] A.J. Viterbi. “Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm”. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.

[109] G. Welch and G. Bishop. “An Introduction to the Kalman Kilter”. *Technical Report 95-041,University of North Carolina at Chapel Hill*, 1995.

[110] C.R Wren, A. Azarbayejani, T. Darrell, and A. Pentland. “Pfinder: Real-Time Tracking of the Human Body”. *In IEEE Transactions Pattern Analysis and Machine Intelligence*, 19, 1998.

[111] B. Wu and R. Nevatia. “Tracking of Multiple, Partially Occluded Humans based on Static Body Part Detection”. *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR’06)*, 2006.

[112] M. Xu, J. Orwell, and G. Jones. “Tracking Football Players with Multiple Cameras”. *In Proc. of IEEE International Conference on Image Processing*, V:2909–2912, 2004.

[113] F. Yan, W.J. Christmas, and J. Kittler. “All Pairs Shortest Path Formulation for Multiple Object Tracking with Application to Tennis Video Analysis”. *In Proc. of BMVA British Machine Vision Conference (BMVC’07)*, 2:6500659, 2007.

[114] D.B Yang, H.H Gonzales-Banos, and L.J Guibas. “Counting People in Crowds with a Real-Time Network of Simple Image Sensors”. *In Proc. of IEEE International Conference on Computer Vision (ICCV’03)*, pages 122–129, 1993.

[115] T. Zhao and R. Nevatia. “Tracking Multiple Humans in a Crowded Enviroment”. *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR’04)*, pages 406–413, 2004.