Stable Multi-Target Tracking in Real-Time Surveillance Video

Ben Benfold Ian Reid

Abstract

The majority of existing pedestrian trackers concentrate on maintaining the identities of targets, however systems for remote biometric analysis or activity recognition in surveillance video often require stable bounding-boxes around pedestrians rather than approximate locations. We present a multi-target tracking system that is designed specifically for the provision of stable and accurate head location estimates. By performing data association over a sliding window of frames, we are able to correct many data association errors and fill in gaps where observations are missed. The approach is multi-threaded and combines asynchronous HOG detections with simultaneous KLT tracking and Markov-Chain Monte-Carlo Data Association (MCMCDA) to provide guaranteed real-time tracking in high definition video. Where previous approaches have used ad-hoc models for data association, we use a more principled approach based on MDL which accurately models the affinity between observations. We demonstrate by qualitative and quantitative evaluation that the system is capable of providing precise location estimates for large crowds of pedestrians in real-time. To facilitate future performance comparisons, we will make a new dataset with hand annotated ground truth head locations publicly available.

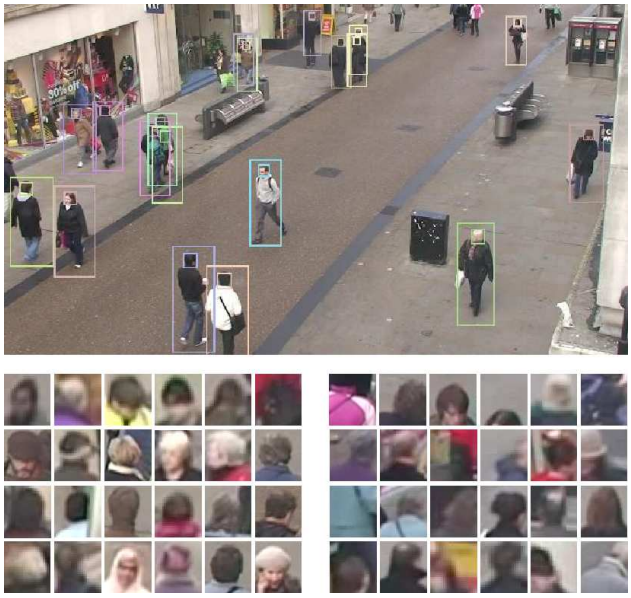

Figure 1. An example of a frame in which we would like to track (top) and sample output from our system (bottom left). The images in the bottom right show the result of a naive approach which applies a fixed offset to a pedestrian detection to estimate the head location, the result of which is badly centred and drifts around as the pedestrian walks.

1. Introduction

(Preprint)[1] The performance of pedestrian tracking systems has steadily increased to the point where attempts are being made to track dense crowds of highly occluded pedestrians. The availability of high resolution surveillance video has opened up new opportunities for exploiting the output of such trackers by using them to produce streams of high resolution pedestrian images. These images can be further analysed to estimate body pose, recognise actions or for passive face recognition.

In this paper we describe a system which is optimised for the provision of accurate pedestrian head locations that are suitable further processing. Our goal is to achieve robustness and high levels of accuracy whilst maintaining real-time (25 fps) performance when tracking multiple pedestrians in high definition surveillance video. Since humans are non-rigid we use the head location as our point of reference because heads are rarely obscured from overhead surveillance cameras and are generally not obscured by clothing. The advantages of tracking heads directly are illustrated in figure 1.

Most recent work on multiple target tracking has focused on appearance based methods, which can be divided into two groups. The first group covers feed-forward systems which use only current and past observations to estimate the current state [7, 1, 2]. The second group covers data association based methods which also use future information to estimate the current state, allowing ambiguities to be more easily resolved at the cost of increased latency [12, 10, 11, 3, 19].

[1] This is a draft version and may contain some minor errors. The full version is copyright and will be available from the IEEE following the CVPR 2011 conference.

The tracking algorithm described in the paper is based around MCMCDA, of which the first instances were developed for tracking a single [4] or fixed number [16] of targets. Oh et al. [15] developed the approach for general multi-target tracking problems to associate sequences of absolute observations into an unknown number of tracks.

Later work developed MCMCDA tracking systems specifically for visual tracking by associating object detections resulting from background subtraction [21] and a boosted Haar classifier cascade [13]. The most recent work [9, 18] further specialises the approach for visual tracking by using not only object detections but also motion estimations or *tracklets* by applying a standard tracking algorithm for a short period of time after each detection. Our method also uses a combination of detections and motion estimates and bears closest resemblance to the work of Ge and Collins [9], however we make a number of improvements.

The first contribution described in the work is the development of a tracking model which accurately represents the error characteristics of the detections and tracklets, allowing the frame-to-frame motion to be estimated with pixel-level accuracy. The resulting location estimates are accurate enough for the generation of stabilised sequences of pedestrian images.

The next contribution involves the treatment of false-positive detections. Previous approaches have assumed that false positives occur as individual uncorrelated detections, however in practice appearance-based detectors often generate repeated false-positive detections in similar locations. We remedy this problem using ideas recently applied to SLAM [6] by creating a separate model for false-positives and combine the identification of false positives with the data association.

The final contribution is a comprehensive evaluation of the tracker on multiple datasets using the standard CLEAR MOT [5] evaluation criteria. Additional experiments assess the performance of the system under various constraints that might affect deployment in different application domains.

Achieving real-time performance remains beyond the reach of most existing tracking-by-detection systems. We note that the detection stage, especially in high definition video, is a bottleneck and most algorithms (notably the successful and popular HOG) cannot run at full framerate even when using GPU acceleration. In addressing this issue we adopt a multi-threaded approach, in which one thread produces asynchronous detections, while another performs local feature based tracking (KLT), a third performs data association and a fourth generates and optimises the output. A key feature of our work is the use of MCMC data association within a temporal sliding window. The advantage of this approach is that at any instant in time the system can report its current best estimate of all target trajectories, but these may change either with more video evidence, or with further iterations. In particular this gives the system the ability to cope with full occlusions for short periods of time.

2. Sliding Window Tracking

2.1. Observations

To make the tracking algorithm robust to false detections, the data association and location estimates are performed by considering all of the data within a sliding window representing the most recent six seconds of video that has been received. We obtain object detections using Dalal and Triggs' Histograms of Oriented Gradients (HOG) [8] based detection algorithm for which we trained a detector using head images rather than full body images. Using a GPU implementation [17] of the HOG detector, detections are received at intervals from approximately 200 milliseconds for PAL resolution video to 1200 milliseconds for 1080p video.

Since detections are received infrequently, motion estimates are necessary to ensure that data associations can be made correctly. We make motion estimates by following corner features with pyramidal Kanade-Lucas-Tomasi (KLT) tracking [14, 20]. To provide robustness against KLT tracking failure, up to four corner features are tracked both forwards and backwards in time from detections for up to s seconds, so between any sequential pair of detections there will be relative location estimates in both directions. In practice we use a value of four seconds for s, which gives good performance without introducing too much latency. KLT tracking was chosen because it is more precise than alternatives such as mean-shift and because tracking multiple corner features provides redundancy against tracking failures. These motion estimations also allow the accurate estimate of head locations between detections.

2.2. Data Association

The purpose of the data association stage is to select a hypothesis H_i which divides the set of detections D into disjoint subsets $T_1, T_2 \dots T_J$ where each subset T_j contains all of the detections corresponding to a single person (see figure 2. Since not every detection that occurs is a true positive, for each T_j we also attempt to infer the type c_j of the corresponding track. We use $c_j = c_{ped}$ to represent the property of T_j being a genuine pedestrian track or $c_j = c_{fp}$ if we believe T_j is a track of false positives, which will be abbreviated to just c_{ped} and c_{fp}. For more general situations, this variable could be extended to represent a number of different moving object types such as cars, bicycles and trees, each of which would have an individual motion model to facilitate classification.

Exhaustively evaluating the space of hypotheses is too slow even for small sets of detections, so we use MCMC

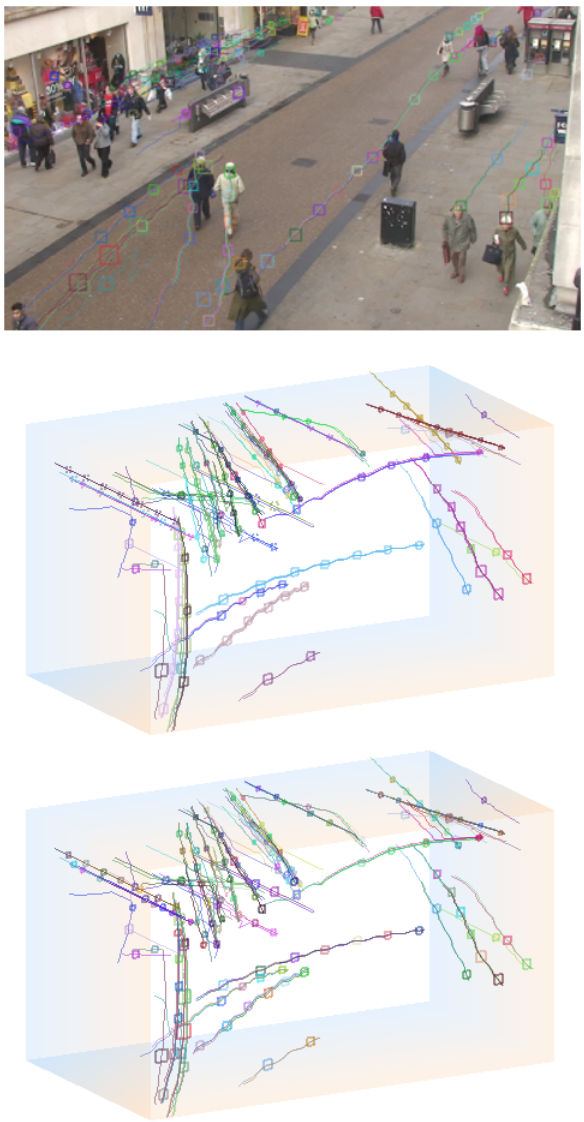

Figure 2. (Requires Colour) Views showing all of the head detections (small rectangles) and the corresponding KLT motion estimates (thin lines) within the sliding window. The colours represent the tracks to which the observations have been assigned. The top image shows the observations projected onto the current frame, the middle plot shows the data association hypothesis that has been made and the bottom image shows the result without data association

sampling to efficiently explore the space of data associations by generating a sequence $H_0, H_1, H_2, \ldots$ of sampled hypotheses. These sampled hypotheses will be distributed according to their relative probabilities which are defined by our likelihood function. The random nature of MCMC helps to prevent the search from becoming stuck in local maxima of the likelihood function.

2.2.1 Likelihood Function

Our likelihood function $p(H_i)$ is proportional to the probability of H_i representing the correct data associations and track types. In previous approaches, the likelihood function has been estimated as a product of a number of terms based on specific properties such as the length of tracks, velocity coherence, spatial overlap and the number of detections considered to be false alarms. We take an approach based on the principles of MDL by attempting to find the hypothesis which allow the most compact representation of the observed detections. The code length L required to encode both the data D and our hypothesis H to a given accuracy is dependent on a corresponding likelihood function:

$$L(D|H) + L(H) = -log(p(D|H_i)p(H_i)) \qquad (1)$$

Although finding the hypothesis which minimises the description length is equivalent to maximising the joint likelihood of the data and the hypothesis, the principles of MDL guide the choice of likelihood function to one which allows observed variables to be encoded efficiently.

First we consider the encoding of the hypothesis H_i, which requires each detection d to be assigned to a track and each track to be given a type label. The track membership is most efficiently encoded when the prior over track membership has a distribution equal to that of the track lengths, resulting in the following prior for H_i:

$$p(H_i) = J! \prod_{T_j \in H_i} \left(\frac{|T_j|}{|D|} \right)^{|T_j|} p(c_j) \qquad (2)$$

where $p(c_j)$ is a prior over the different track types and the notation $|D|$ is used to denote the cardinality of the set D. The factor of $J!$ arises because the ordering of the subsets is not important, so the first detection in any track may be encoded with any of up to J identifiers which have not already been used.

Detections genuinely from the same track are expected to be highly correlated so can be efficiently encoded in terms of one another once divided into tracks. The improvement in encoding efficiency will almost always save more information than is required to store the hypothesis. Next we break down the likelihood function into components representing each track. Let d_n^j be the nth detection in a track T_j, where the index n indicates only the order within the track

$$p(D|H_i) = \prod_{T_j \in H_i} \left[p(d_1^j|c_j) \prod_{d_n^j \in T_j \setminus d_1^j} p(d_n^j|d_{n-1}^j, c_j) \right] \qquad (3)$$

For each detection, we would like to encode the scale of the detection s_n, the location x_n within the frame, and an approximation to the KLT motion m_n. To ensure equivalent behaviour over different scales, the location accuracy is measured relative to the size of the object, so the units of x_n are multiples of s_n rather than pixels. Ideally we would consider the coding of the KLT motion estimates too, but due to the quantity of data this would have a negative impact on performance. Since the magnitude of the KLT motion is important for distinguishing between true positive and false positive detections, we instead approximate the motion by building a histogram m_n from the magnitude of every frame-to-frame motion estimate originating from detection n. The likelihood functions for individual detections are then be broken down into components representing these observed properties:

$$p(d_1^j|c_j) = p(s_1)p(x_1)p(m_1|c_j) \tag{4}$$

$$p(d_n^j|d_{n-1}^j, c_j) = p(s_n|s_{n-1})p(x_n|x_{n-1}, c_j)p(m_n|c_j) \tag{5}$$

Variables upon which the probabilities are conditional have been omitted where independence is assumed.

Detection Scales The scale of the first detection in each track cannot be encoded in terms of any preceding detection, so a global prior log-normal distribution with mean μ_p and variance σ_p^2 is assumed:

$$\ln s_n \sim N(\mu_p, \sigma_p^2) \tag{6}$$

The scales for the following detections in the track can then be encoded more efficiently in terms of the previous scale:

$$\ln \frac{s_n}{s_{n-1}}|c_{ped} \sim N(0, \delta_t \sigma_{sp}^2) \tag{7}$$

$$\ln \frac{s_n}{s_{n-1}}|c_{fp} \sim N(0, \delta_t \sigma_{sf}^2) \tag{8}$$

where δ_t is the time difference between the frames in which the detections were made.

Image Location A similar approach is used when considering the optimal method for encoding the image location. It is assumed that the locations of both pedestrians and false-positives are uniformly distributed around the image, so the probability density of x_n depends on the image area a relative to the object size in pixels:

$$p(x_n) = \frac{a}{s_k^2} \tag{9}$$

For subsequent detections, the locations can be better explained in terms of the preceding detections, however the way in which we do this depends on the track type c_j. For genuine pedestrians, we first make a prediction based on a constant velocity model:

$$x_p = x_{n-1} + \delta_t v_p \tag{10}$$

$$\Sigma_p = \delta_t \Sigma_v \tag{11}$$

the velocity estimate v_p comes from the result of the KLT tracking in the frames immediately before and after the detection, by which point it is unlikely to have failed. The error in the velocity due to unknown accelerations is modelled by the parameter Σ_v. Whilst the constant velocity model is an improvement on the uniform prior, the error Σ_p is still large partly due to the cyclic gait motion and also because detections are infrequent and humans often change direction when in crowds. The full KLT motion estimates generally provide much more accurate predictions, so for each KLT motion estimate y we calculate a posterior distribution over locations using a calculation equivalent to the update step of a Kalman filter:

$$x_y = x_{n-1} + \Sigma_p(\Sigma_p + \delta_t \Sigma_{klt})^{-1}(x_{n-1} + y - x_p) \tag{12}$$

$$\Sigma_y = (I - \Sigma_p(\Sigma_p + \delta_t \Sigma_{klt})^{-1})\Sigma_p \tag{13}$$

The parameter Σ_{klt} represents the rate at which KLT feature tracking accumulates random error and δ_t is the time difference between detections d_n and d_{n-1}. The possibility that a tracked KLT feature fails completely is modelled using the parameter α, where $(1-\alpha)^{\delta_t}$ is the probability of failure after tracking for δ_t seconds. The detection location is then encoded using a mixture of the prior and posterior distributions:

$$x_n|x_{n-1}, \mathbf{y}, c_{ped} \sim \alpha^{\delta_t} \frac{1}{|\mathbf{y}|} \sum_{y \in \mathbf{y}} N(x_y, \Sigma_y + 2\Sigma_d) + (1 - \alpha^{\delta_t}) N(x_p, \Sigma_p + 2\Sigma_d) \tag{14}$$

In the event that $\mathbf{y}$ is empty, α is set to 0 and the first term is omitted. The additional uncertainty of $2\Sigma_d$ is included to model the error in the two detection locations. Tracks consisting of repeated false-positives are usually caused by static objects in the background so are assumed to be stationary:

$$x_n|x_{n-1}, \mathbf{y}, c_{fp} \sim N(x_{n-1}, 2\Sigma_d) \tag{15}$$

Motion Magnitude The last observation considered is the motion magnitude histogram. This is included only to help distinguish between false positives which are expected to have no movement and true positives which are expected to move at least a small amount, so the histogram has just four bins with boundaries representing movement of $\frac{1}{8}$,$\frac{1}{4}$

and $\frac{1}{2}$ pixels per frame. The histograms are expected to conform to one of two multinomial distributions depending on the type of track:

$$m_n|c_{ped} \sim Mult(m_{ped}) \quad (16)$$
$$m_n|c_{fp} \sim Mult(m_{fp}) \quad (17)$$

Throughout this section the probability of each detection being in a track has been calculated by considering only the immediately preceding and immediately following detections. This first-order approximation was made to ensure that the MCMC data association can be performed efficiently, with each proposal requiring only a minimal calculation. The optimisation of the joint probabilities is deferred to section 2.3.

2.2.2 Sampling

There are three types of move which can be made during the sampling process, the first two moves effect the state of the data association and the third has the potential to change the type of a track. The first type of move involves randomly selecting one detection and one track, and proposing that the chosen detection be moved to the chosen track, as illustrated in figure 3. In the event that the track already contains a detection from the same frame, both are swapped in the proposal. In the second type of move we choose two tracks and a random time within the sliding window and propose that all of the detections occurring after the time in both of the tracks are swapped with those in the other.

To calculate the likelihood $p(H^*)$ of the first proposal requires at most four evaluations of equation 5 and the second requires no more than two. The third move type, in which a change of track type is proposed, requires the probability of every detection in a single randomly chosen track to be re-evaluated. Fortunately this third move type depends on just one track so does not need to be attempted as frequently.

The Metropolis-Hastings acceptance function defines the likelihood with which the proposal should be accepted:

$$p(H_{i+1} \Leftarrow H^*) = \min\left(\frac{p(H^*)q(H_i|H^*)}{p(H_i)q(H^*|H_i)}, 1\right) \quad (18)$$

In most cases the proposal density q will be the same for both the forward and the reverse move, however there are some cases where is is not. Random tracks for proposals are drawn from the set of existing tracks plus one additional empty track. This empty track allows a new track to be created, and similarly either of the two moves could leave one of the tracks empty in which case it is destroyed. Since only one empty track is retained, the creation or destruction of a track effects the probability of the move being made.

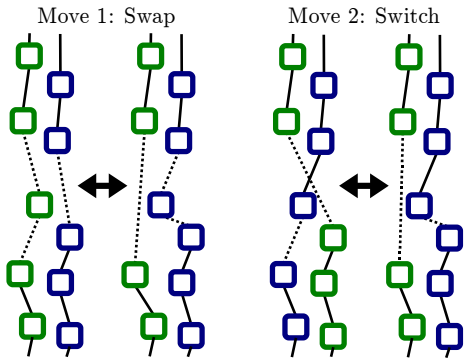

Figure 3. Examples of the first two types of move used for MCMCDA. Only the probabilities for pairwise associations with dotted lines need to be recalculated when each move is proposed.

Although Metropolis-Hastings is good at overcoming local maxima of the likelihood function, we prefer stable output rather than samples. To obtain stable output we keep track of the most likely hypothesis since observations were last received and output the local maximum, which is found by only accepting proposals that are more likely than the current hypothesis for a short period of time.

2.2.3 Parameter Estimation

Some of the model parameters such as the detector covariance are likely to depend on characteristics of the particular video sequence such as the level of image noise and blur. In our system these are learned automatically using an approach based on that of Ge and Collins [9] by interleaving the MCMCDA sampling with additional Metropolis-Hastings updates of the parameters. Provided the parameters are initialised to values allowing some tracks to be correctly associated, both the parameter samples and the data association converge to a good maximum of the likelihood function. Parameter updates take considerably longer than data association updates because the log-likelihood must be recalculated for all of the data. In a live system the parameters can be learned online over an hour or two. However, since most datasets are too short for this, we slow down the video used for training so that there is enough time to learn the parameters.

2.3. Output Generation

The final stage is to generate estimates for the object location in each frame. First we estimate the true image locations $\hat{x}_n$ for all of the detections in each track.

$$p(\hat{T}_j) = \prod_{1<n\leq N} p(\hat{x}_n|\hat{x}_{n-1}, \mathbf{y}, c_{ped})p(\hat{x}_n|x_n) \quad (19)$$
$$\hat{x}_n|x_n \sim N(x_n, \Sigma_d) \quad (20)$$

The term $p(\hat{x}_n|\hat{x}_{n-1}, \mathbf{y}, c_{ped})$ is equivalent to equation 14 but without the $2\Sigma_d$ terms. Since multiple KLT estimates result in multimodal probability distributions, we again optimise using Metropolis-Hastings sampling.

Since detection do not occur in every frame, the location estimates for the other frames are made by interpolating between detections. Interpolations are made by averaging each of the relevant KLT motion estimates weighted by the corresponding contribution to the mixture in equation 14.

3. Evaluation

The purpose of our system is to provide stable head location estimates in surveillance video, however there are no standard datasets for evaluating this so we use our own video dataset of a busy town centre street. The video is high definition (1920x1080/25fps) and has ground truth consisting of 71500 hand labelled head locations, with an average of sixteen people visible at any time. Both the ground truth for this sequence and our tracking output will be made publicly available to encourage future comparisons.

Four metrics are used to evaluate the tracking performance. The Multiple Object Tracking Precision (MOTP) measures the precision with which objects are located using the intersection of the estimated region with the ground truth region. Multiple Object Tracking Accuracy (MOTA) is a combined measure which takes into account false positives, false negatives and identity switches (see [5] for details). The detection detection precision and recall are also included to enable comparisons with other work. Since head regions are considerably smaller than full body boxes, any error in the location estimates has a much more significant impact on the performance measures than for the full body regions. For this reason the two should not be directly compared, however to allow some comparison to be made with full-body trackers, we also calculate the performance measures using full-body regions that are estimates from the head locations using the camera calibration parameters and a known ground plane. All experiments were performed on a desktop computer with 2.4GHz quad-core CPU with GPU accelerated HOG detections and in real-time unless otherwise stated.

The results for the town centre video are shown in table 1. Since there are no similar head tracking results to compare with, baseline results from raw HOG head and full body detections are included for comparison. We also examine the effects of adjusting the latency between when frames are received and when output is generated for them (figure 4) because this is relevant for many practical applications. Some insight into the cause of most tracking failures can also be gained from figure 5, which shows how tracking at lower speeds affects the performance. The result is that although more frequent detections increase the recall, the precision drops because there are more false positives.

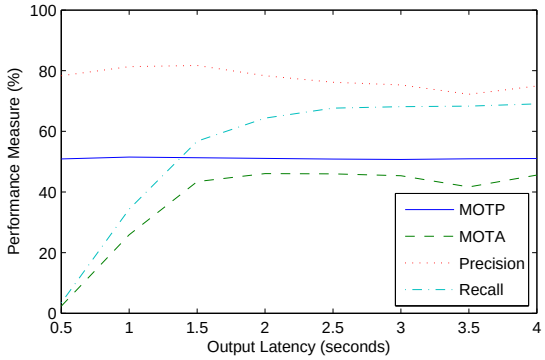

Figure 4. Reducing the latency introduced between frames arriving and their output being generated causes the recall to decrease. Performance measure are for head regions.

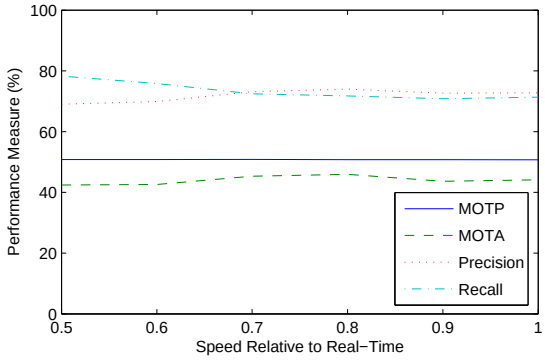

Figure 5. A graph showing the affect of slowing down the system so that more time is available for processing the video. Performance measures are for head regions.

These false positives are the result of incorrect head detections on other body parts such as shoulders or on bags and often occur repeatedly in the same place as the pedestrian moves.

Whilst the system we describe was intended for the purpose of obtaining stable image streams, we also demonstrate the general tracking performance by performing a quantitative analysis on a standard test video from the i-Lids AVSS 2007 dataset. The video is of a train station platform, has a resolution of 720 x 576 pixels at 25 fps and has 9718 labelled ground truth regions. The ground truth is for whole pedestrians and we only track heads, so the full body regions were estimated using an approximate camera calibration with a ground plane assumption. Since the video includes distant people that are too small for the head detector to detect, alternate detections were performed with the standard full body HOG detector with a fixed offset to estimate the head location. A separate detection covariance parameter was learned for full body detections. The results are shown in table 2.

Method	Head Regions				Body Regions			
	MOTP	MOTA	Prec	Rec	MOTP	MOTA	Prec	Rec
Our tracking	50.8%	45.4%	73.8%	71.0%	80.3%	61.3%	82.0%	79.0%
HOG head detections	45.8%	-	35.0%	52.7%	76.1%	-	49.4%	74.5%
HOG body detections	44.3%	-	44.7%	39.3%	72.7%	-	82.4%	72.3%

Table 1. Tracking performance on the town centre sequence for our tracking system and baseline estimates using raw HOG head and full body detections. Body regions from the HOG detector were converted to head regions using a fixed offset and head regions were converted to body regions using the camera calibration. MOTA takes into account identity switches so cannot be calculated without data association. The HOG detectors were applied to every video frame, which is approximately ten times too slow to run in real-time.

Method	MOTP	MOTA	Prec	Rec
Ours	73.6%	59.9%	80.3%	82.0%
Breitenstein et al. [7]	67.0%	78.1%	-	83.6%
Stalder et al. [19]	-	-	89.4%	53.3%

Table 2. Tracking performance for full body regions on the i-Lids AB Easy sequence using both the CLEAR metrics and standard detection precision and recall.

In addition to these datasets, we also show qualitative results on the PETS 2007 videos, for which we do not have ground truth data, to demonstrate the ability of the system to cope with dense crowds of people. Figure 6 shows sample frames from all three sequences along with cropped head regions to demonstrate the stability of the tracking.

4. Conclusions

We have described and demonstrated a scalable real-time system for obtaining stable head images from pedestrians high definition surveillance video. The use of MCMCDA makes the system robust to occlusions and allows false positive detections in the background to be identified and removed. The system has many potential applications in activity recognition and remote biometric analysis or could be used to provide close-up head images for a human observer. The experiments performed show that our efficient approach provides general tracking performance comparable to that of similar systems, whilst being advantageous in terms of speed and tracker stability.

References

[1] I. Ali and M. Dailey. Multiple human tracking in high-density crowds. In *Advanced Concepts for Intelligent Vision Systems*, volume 5807 of *LNCS*, pages 540–549. 2009. 1

[2] B. Benfold and I. Reid. Guiding visual surveillance by tracking human attention. In *BMVC*, September 2009. 1

[3] J. Berclaz, F. Fleuret, and P. Fua. Multiple object tracking using flow linear programming. In *Winter-PETS*, December 2009. 1

[4] N. Bergman and A. Doucet. Markov chain monte carlo data association for target tracking. In *ASSP*, volume 2, pages II705 –II708, 2000. 2

[5] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. In *EURASIP JIVP*. 2008. 2, 6

[6] C. Bibby and I. Reid. Simultaneous localisation and mapping in dynamic environments (SLAMIDE) with reversible data association. In *RSS*, 2007. 2

[7] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *ICCV*, pages 1515–1522, September 2009. 1, 7

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 2, pages 886–893, June 2005. 2

[9] W. Ge and R. T. Collins. Multi-target data association by tracklets with unsupervised parameter estimation. In *BMVC*, 2008. 2, 5

[10] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *ECCV (2)*, volume 5303 of *LNCS*, pages 788–801. Springer, October 2008. 1

[11] B. Leibe, K. Schindler, and L. J. V. Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV*, pages 1–8. IEEE, October 2007. 1

[12] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR*, pages 2953–2960. IEEE, June 2009. 1

[13] J. Liu, X. Tong, W. Li, T. Wang, Y. Zhang, H. Wang, B. Yang, L. Sun, and S. Yang. Automatic player detection, labeling and tracking in broadcast soccer video. In *BMVC*, 2007. 2

[14] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, pages 674–679. William Kaufmann, 1981. 2

[15] S. Oh, S. Russell, and S. Sastry. Markov chain monte carlo data association for general multiple-target tracking problems. In *ICDC*, 2004. 2

[16] H. Pasula, S. J. Russell, M. Ostland, and Y. Ritov. Tracking many objects with many sensors. In *IJCAI*, pages 1160–1171. Morgan Kaufmann, 1999. 2

[17] V. Prisacariu and I. Reid. fastHOG - a real-time GPU implementation of HOG. Technical Report 2310/09, University of Oxford, 2009. 2

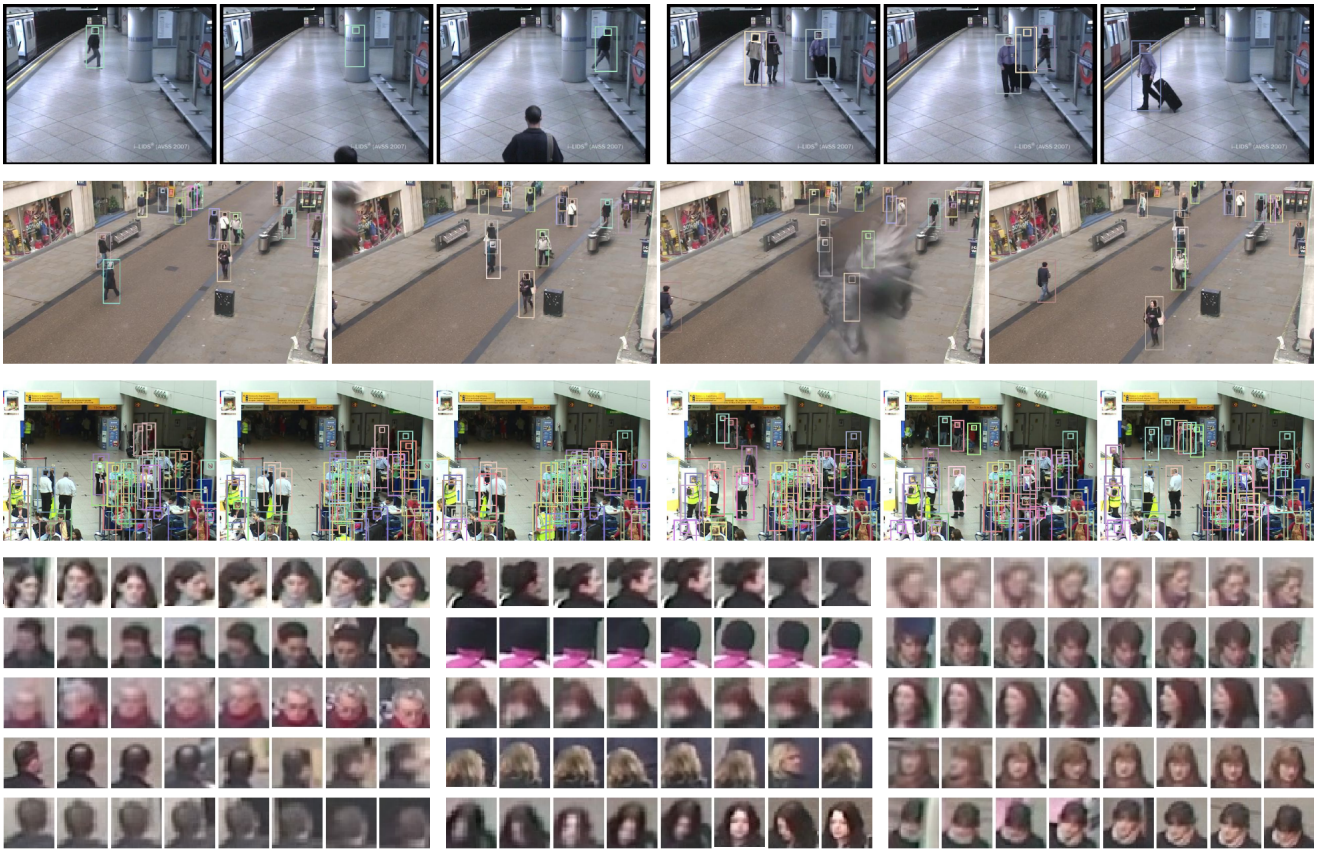

Figure 6. Sample video frames from the i-Lids sequence (first row), town centre (second row) and PETS 2007 (third row). The first two video sequences demonstrate the ability of the system to deal with temporary occlusions, caused by the pillar in the i-Lids video and by a pigeon in the town centre video. The bottom rows show example of the stable head sequences that result from the tracking.

[18] B. Song, T.-Y. Jeng, E. Staudt, and A. K. Roy-Chowdhury. A stochastic graph evolution framework for robust multi-target tracking. In *ECCV*, volume 6311 of *LNCS*, pages 605–619. Springer, 2010. 2

[19] S. Stalder, H. Grabner, and L. J. V. Gool. Cascaded confidence filtering for improved tracking-by-detection. In *ECCV (1)*, volume 6311 of *LNCS*, pages 369–382. Springer, September 2010. 1, 7

[20] C. Tomasi and T. Kanade. Detection and Tracking of Point Features. Technical Report CMU-CS-91-132, Carnegie Mellon University, Apr. 1991. 2

[21] Q. Yu, G. G. Medioni, and I. Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *CVPR*. IEEE, 2007. 2