CROWD BEHAVIOUR ANALYSIS USING HISTOGRAMS OF MOTION DIRECTION

Hannah M. Dee and Alice Caplier

GIPSA lab, Grenoble INP,
Domaine Universitaire BP 46 38402
Saint Martin d'Hères cedex
hannah.dee@gipsa-lab.grenoble-inp.fr,alice.caplier@gipsa-lab.grenoble-inp.fr

ABSTRACT

A practical system for the automated analysis of crowded scenes will have to deal with multiple occlusions and tracking failures, in a context in which the cameras may move at any time to point in any direction, at any level of zoom. This paper presents a prototype component of such a system. Much work in crowd modelling assumes that the camera will be static for extended periods of time and that a model of the scene can therefore be learned; we do not make this assumption and instead build a simple representation of motion patterns that is applicable across different views and which learns motion scale rapidly. Our representation is based upon histograms of motion direction alongside an indication of motion speed. These can be used for detecting frames in which behaviour differs from the training set, and also for localisation of where in the image these anomalous events occur. We evaluate this work against five event-detection scenarios from the public PETS2009 crowd behaviour dataset.

Index Terms— Machine vision, Site security monitoring, Video processing

1. INTRODUCTION

The automated analysis of crowd behaviour is a growing field of research within computer vision, with applications to crowd safety, event detection, and security. The problems of crowd analysis have been well described elsewhere [1] and include multiple occlusions, self-occlusions, and motion in unpredictable directions. These issues often result in the failure of tracking systems, particularly those based upon traditional background subtraction methodologies.

The problems of modern-day surveillance systems are also well known, in which a small number of operatives are tasked with monitoring a large number of cameras – often one operative to more than 100 video feeds. Thus systems have to deal with camera feeds which are primarily static, but which can move at any time. This paper presents a prototype system for analysing crowd behaviour from such streams, consisting of two components: a rapid and robust motion scale estimation system, and a crowd behaviour modelling component based upon histograms of motion direction (HMDs). The crowd behaviour modelling element shares some characteristics with earlier work [2], in which static scenes are segmented spatially through the analysis of HMDs. The current paper represents both an extension and a simplification of this approach: the earlier work disregarded temporal information and built a representation which was purely spatial using multiple HMDs to learn across many scenes and over extended periods of time (hours). In contrast, the current paper describes a system which uses HMDs learned from a small training set of "normal" behaviour to determine whether particular crowd behaviour videos contain unusual events. Using the scale estimation system, we are able to incorporate ideas of motion scale, which were unavailable to the earlier method, and by learning a simple representation per training video we are able to learn rapidly (within seconds).

The organisation of this paper is as follows: Section 2 provides an overview of related work; Section 3 describes the motion scale estimation module; Section 4 the crowd behaviour analysis system; and Section 5 describes evaluation of the system on a public crowd analysis dataset. Finally conclusions are presented in Section 6.

2. RELATED WORK

Approaches to event detection and behaviour modelling in crowded scenes often involve modelling the way in which a crowd moves over extended periods of time: optical flow is used in [3] to detect crowd motion and then an SOM models the way in which this motion is distributed in a scene. In [4] "floor fields" are learned which can describe the way a crowd's motion interacts with the scene geography; [5] uses particle dynamics to model the way that the elements of a crowd interact; and in [6] "supertracks" are recovered from instantaneous motion patterns through a sink-seeking procedure. The motion of corner features is modeled with Gaussian mixtures in [7].

A second approach models the motion patterns found in the scene without tying a particular observed motion to a loca-

tion in ground- or image- plane. These systems either model the scene as a whole, or divide the scene into regions and learn the behaviour in each region. [8] use background subtraction and corner detection to detect the motion of a crowd and then apply *a-priori* criteria (e.g. speed constraints, motion variance) to detect events. In [9] a similar approach to feature extraction is coupled with a mixture of von Mises (sometimes called "circular normal") distributions to model the way in which direction varies within the scene. Interestingly, [10] mention the use of a histogram of motion direction but do not show how these could be used for automated event detection.

3. MOTION SCALE ESTIMATION

The aim of our motion scale estimation component is to rapidly provide a rough estimate of the size of objects seen from a particular camera view. By making the simplifying assumptions that the scene contains people and that perspective effects are not strong, we estimate motion scale from output of either face or person detectors. Given a static camera, we apply a HOG based pedestrian detector [11] and a Viola-Jones [12] face detector. Unlike previous work in this area which aims to model details of scene geography including perspective effects (for example, [13]) we do not attempt to build a detailed or accurate model of object size but merely to determine the rough scale of motion. The conversion between person detections and estimated face location is based upon the mean size of a face within the bounding box returned by the HOG detector: on average, the face takes up 25% of the bounding box width and starts 18% from the top of the box. The detections from these systems are stored and when a sufficient number of detections has been reached (from either detector), we take the average face size $\bar{w}$ and use this as a normalisation factor.

This scale estimation was evaluated against the PETS2009 dataset, in which 100 faces have been hand-labelled in each of the four views. Figure 1 shows the influence of the number of detections against the mean square error when comparing predicted face width against the ground truth labellings. This shows that with 60 detections the worst mean square error is just over 16 pixels-squared. We have also performed informal evaluation against a moving-camera CCTV dataset and find that this works well, adapting to camera motion within 10-30 frames.

4. CROWD BEHAVIOUR ANALYSIS AND EVENT DETECTION

The input video is first passed to a "KLT" feature tracker [14] which we reinitialise every second. This provides us with short *tracklets* which are long enough to provide a robust indication of object motion, but short enough to avoid many of the occlusion-related problems that crowded areas can cause for systems based upon longer tracks. A visualisation of the

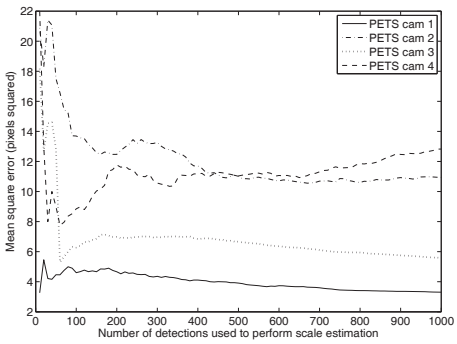

Fig. 1. The mean square error for face-width estimates over four camera views of the PETS2009 dataset

Fig. 2. KLT tracklets for one of the test sets from PETS2009. These provide a robust indication of motion within the scene

KLT tracks for one of the test sets is shown in Figure 2. Each tracklet is then quantized into 8 cardinal directions (up, up-left, left, ...) based upon the angle between start and end points, and the distance travelled over the course of the tracklet is normalised through multiplication by a scale factor of $1/\bar{w}$ to give speed s in approximately consistent units across scenes. Accumulation of tracklet counts in each of these 8 quantized directions form our HMDs

4.1. Event detection

To use HMDs with scale estimation for supervised event detection, we calculate whole scene HMDs H_t, mean tracklet speed $\bar{s}_t$, and standard deviation σ_t in tracklet speed from a training video or videos assumed to contain only "normal" behaviour. It is worth noting here that the representation we acquire from this training is very compact; each video is represented by 8 integers (the directional information) and two floating point numbers (the motion scale information).

Having learned the HMD H_t, average speed $\bar{s}_t$ and the

1: Running Crowd enters walking, at frame 50-105 runs out; re-enters walking, from frame 180-end runs out again

2: Loitering Crowd stands around in centre of scene, moves a little at 98-135, moves a little again at 175-185.

3: Dispersal Crowd starts in centre of scene then disperses in all directions from frame 70-end.

4: Dispersal Movement from edge of scene to centre, rapid dispersal at frame 320-end. Camera shake at frame 250.

5: Formation Movement from edge of scene to centre, from frame 80 crowd walks out of view.

Fig. 3. Description of test sets from PETS2009

standard deviation of the speed σ_t from the training videos T, in testing we compare the query video HMDs H_q and average speed $\bar{s_q}$ with these using Equation 1. Rather than work at the level of the entire image, we divide the scene into M by M pixel squares, and calculate HMDs for each of these.

$$d(T,Q) = \sum_i (1 - c(H_t, H_{qi})) + \lambda \frac{\bar{s_{qi}} - \bar{s_t}}{\sigma_t} \qquad (1)$$

Where $c(H_t, H_q)$ is the correlation distance between the two histograms, and λ is a scale factor chosen to balance the contributions of histogram distance and speed difference. In our experiments, $\lambda = 0.04$. Within the query video, we sum the HMDs between each square and those from the training video(s) to get an indication of how the input scene compares to the training video(s) on a global scale (*event detection*), and then compare each individual M by M square's measurements to the training video(s) to determine which scene regions events may be occurring in (*event localisation*). If we use more than one training video, we compare query HMDs and speeds to each of the training datasets and take the minimum distance.

We also consider the use of relative HMDs (RHMDs) in which the bins have been shifted so that the largest bin entry is always in the first bin. This can be thought of as "rotating" the motion histogram so that the greatest quantity of motion is in the "up" direction. This provides a representation of the way in which motion direction varies, without actually representing the actual directions of motion. These RHMDs (whilst losing some information about the motion in a scene) have the advantage of allowing us to compare broad types of motion across camera views. Due to space constraints, all results presented in this paper are for RHMDs.

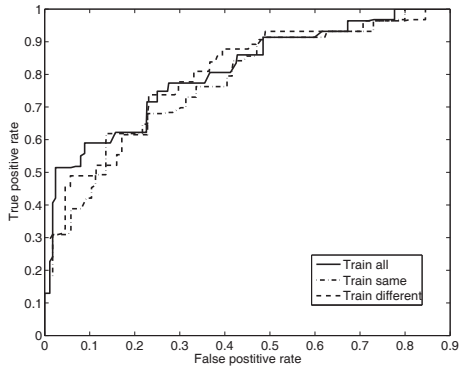

Fig. 4. ROC curves showing the performance on each training set - All, Same, and Different.

5. EVALUATION

This work has been evaluated against all five "event detection" videos from PETS2009. These short videos represent different types of simulated crowd events, with each scenario recorded from 4 different camera viewpoints. The test datasets are summarised in Figure 3.

Training has been carried out on the *regular flow* training set from the same dataset, which is taken from the same 4 camera viewpoints and shows a large group of people walking in the outdoor scene. We present results for three training scenarios. "All" refers to training upon all viewpoints. "Same" refers to training on the same viewpoint as the test set, which can be seen as making the assumption of a static camera. "Different" refers to training in which we exclude the test viewpoint from the training set. This final training scenario shows the applicability of our technique to situations with moving cameras, as the test view has not been seen in training. Figure 4 shows ROC curves for event detection. The similarity between the curves in this figure demonstrates the robustness of this method to viewpoint changes.

It is clear from these results that we achieve better performance on some test sets than others. In particular, test set 5 causes problems for our technique, as the event to be detected occurs at the very start of the sequence (whilst our motion estimation component stabilises). A further point to make is that using frame-wise ground truth will introduce additional errors for our system. The results for each scene are given in Table 1, based upon thresholding of $d(T,Q)$ at 3.5 across all datasets. The final column of this table ("Allowance") is calculated as ((Number of events to detect * length of tracklets)/total number of frames in video) and is therefore an indication of the expected error introduced by the temporal granularity.

For the localisation of events of interest, we use the distance from the training HMDs to each M by M square $d(T,Q_i)$ as a measure of the way in which that particular

	All		Different		
Dataset	FPR	TPR	FPR	TPR	Allowance
1	0.04	0.80	0.17	1.00	0.07
2	0.27	0.88	0.46	0.88	0.05
3	0.11	0.75	0.50	1.00	0.06
4	0.34	0.89	0.54	0.89	0.05
5	0.60	1.00	0.64	1.00	0.07

Table 1. Summary of event-detection results. The "allowance" has *not* been applied to the false positive and true positive rate figures in the table, and is included merely to demonstrate the size of detrimental effect we expect from comparing frame-based ground truth to a system which works with 1-second tracklets.

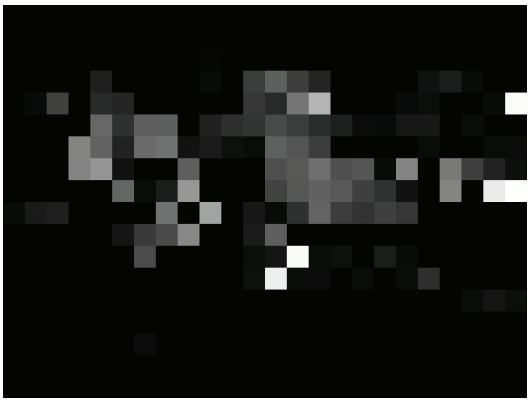

Fig. 5. Event localisation; in this image M is 32 and each square in the image is highlighted based upon $d(T, Q_i)$. Paler squares have a greater value. This frame is pictured in Figure 2

scene region differs from the expected behaviour over the whole scene. Because of the sparseness of the tracklet representation, we accumulate each square's HMD over a two second rather than a one second window, making the behaviour localisation more robust spatially but introducing extra time-lag. Figure 5 shows a visualisation of the localisation results.

6. CONCLUSIONS

In this paper we have presented a method for detecting events in crowded scenes which requires little training data. It has been demonstrated to work well at detecting a variety of events in a publically available crowd dataset.

One limitation of this approach is the assumption that the motion in the scene will be consistent; whilst this assumption is violated in some scenarios it is nonetheless a reasonable one in many surveillance applications. Future work will involve improving the scale estimation component in order to increase performance across different camera views, and extending the system to work in situations with moving cameras.

7. REFERENCES

[1] B. Zhan, N. D. Monekosso, P. Remagnino, S. A. Velastin, and L. Xu, "Crowd analysis: a survey," *Machine Vision and Applications*, vol. 19(5-6), pp. 345–357, 2008.

[2] H. M. Dee, D. C. Hogg, and A. G. Cohn, "Scene modelling and classification using learned spatial relations," in *Conference on Spatial Information Theory*, 2009, vol. 5756.

[3] B. Zhan, P. Remagnino, N. D. Monekosso, and S. A. Velastin, "Self-organizing maps for the automatic interpretation of crowd dynamics," in *Proc. ISVC*, 2008.

[4] S. Ali and M. Shah, "Floor fields for tracking in high density crowd scenes," in *Proc. ECCV*, 2008.

[5] S. Ali and M. Shah, "A Lagrangian particle dynamics approach for crowd flow segmentation and stability analysis," in *Proc. CVPR*, 2007.

[6] M. Hu, S. Ali, and M. Shah, "Detecting global motion patterns in complex videos," in *Proc. ICPR*, 2008.

[7] A. Utasi, A. Kiss, and T. Szirányi, "Statistical filters for crowd image analysis," in *PETS workshop at CVPR*, 2009.

[8] N. Ihaddadene and C. Djeraba, "Real-time crowd motion analysis," in *Proc. ICPR*, 2008.

[9] Y. Benabbas, N. Ihaddadene, and C. Djeraba, "Global analysis of motion vectors for event detection in crowd scenes," in *PETS workshop at CVPR*, 2009.

[10] A. Albiol, M. J. Silla, A. Albiol, and J. M. Mossi, "Video analysis using corner motion statistics," in *PETS workshop at CVPR*, 2009.

[11] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," *Proc. CVPR*, vol. 1, 2005.

[12] P. Viola and M. Jones, "Robust real-time object detection," in *Second Int. Workshop on Statistical and Computational theories of Vision - Modeling, Learning, Computing and Sampling*, 2001.

[13] M. D. Breitenstein, E. Sommerlade, B. Leibe, L. Van Gool, and I. Reid, "Probabilistic parameter selection for learning scene structure from video," in *British Machine Vision Conference*, 2008.

[14] J. Shi and C. Tomasi, "Good features to track," in *Proc. CVPR*, 1994.