

On the feasibility of using a cognitive model to filter surveillance data

H. M. Dee and D. C. Hogg
School of Computing
University of Leeds
Leeds, LS2 9JT, UK.

Abstract

This paper describes a novel approach to the problem of automated visual surveillance. The authors have extended an existing algorithm which uses a cognitive model of navigation to explain behaviour in a surveillance setting. We then take this cognitive model and apply it to the problem of filtering surveillance data: typically, a surveillance or CCTV installation will have a limited number of operatives monitoring a large number of cameras. The proposed system filters upon inexplicability scores, on the grounds that those trajectories which we can explain in terms of simple goals are exactly those trajectories which are uninteresting: it is only those we cannot simply explain which are worth attending to. Initial results are promising, with over 50% of uninteresting trajectories being excluded.

1. Introduction

Current real-world surveillance systems are labour intensive and often ineffectual. Full time monitoring of large numbers of cameras is prohibitively expensive so only a subset of cameras are ever actually watched. Those cameras which are unwatched can only be used for reactive policing, that is, stored images can be used as evidence after a crime is known to have occurred. If the attention of CCTV operatives could be directed to those cameras where something might be happening, the system could be used to direct more proactive policing.

The question of which cameras to watch is a difficult one to answer. Many existing systems involve the operators themselves selecting which cameras to monitor. This leaves the system open to abuse and discrimination in a way that has attracted the ire of human rights and anti-surveillance groups. Studies show that CCTV operatives deciding which cameras to monitor are guided less by the behaviour of the people in the scene and more by their appearance. This is probably inevitable, given the snap decisions which have to be made, with a small number of operatives monitoring several hundred cameras. If the operative only has a few seconds in which to make their judgement, all they can really call upon are static cues such as appearance. A further

problem with CCTV operators is the obvious one of boredom: in the vast majority of surveillance situations, nothing happens[11].

The “holy grail” of automated surveillance is a system which can monitor hundreds of cameras at the same time and draw the attention of operatives to those few cameras where a crime might be being committed. To do this, we would need some form of unusual behaviour detector. A number of systems have been constructed which go some way towards addressing this problem (see, for example, [6, 9, 5]). Many of these earlier systems have been closely tied to the geography of the scene which can limit their application to changing environments. What these previous approaches have in common is that they ignore the underlying *intentional* nature of the agents within the scene – indeed, they are often referred to as *objects*, rather than *agents*. We argue that if the behaviour exhibited by an agent is explicable in terms of a simple model of goal-directed behaviour, then that agent be ignored.

The algorithm presented here is an extension of that presented in [3, 4], applied in a novel way. The paper begins with a description of the basic algorithm (focussing upon the extensions) and its underlying theory for completeness, and goes on to consider the use of such an algorithm to provide a filter on surveillance data.

2 Measuring intentionality

The hypothesis that this paper sets out to investigate is whether or not a measure of *intentionality* can be used as a filter on surveillance videos. This hypothesis stems from the observation that when watching surveillance videos, one of the questions we ask ourselves is “what are they doing?”. This question can be re-cast for each agent within the scene as “what is that agent’s goal?” In the context of visual surveillance, by *goal* we are referring to a geographical goal such as a door, or a parked car. As we hypothesise about which of the geographical goals in the scene are the goals of the agent, we formulate an *explanation* of that agent’s behaviour. If we can explain away their actions, in terms of one of the known geographical goals in the scene, then their

trajectory can be ignored: they are simply *walking towards a particular exit or car*.

In a featureless scene - one without obstacles - we would expect people to cross the scene in a straight line in the direction of one of the scene exits. In a complicated scene such as a car-park or a pedestrianised area, there are obstacles which affect the paths people choose to take through the scene. There are three aspects of the scene which need to be modelled or captured for the intentionality of the agents to be in any way measurable:

- The location and direction of travel of any agents moving around within the scene
- A model describing the location and extent of the exits within the scene, as these are the goals which will be used to explain behaviour
- A model describing the location and extent of any obstacles within the scene

The question of whether or not a particular agent is behaving explicitly for a particular scene becomes a question of whether, *given a particular arrangement of obstacles* that agent might be navigating around those obstacles to one of the scene’s goals.

The two test scenes used in this paper are: the “PETS2004” scene¹ filmed indoors, in a foyer, with actors; and the “car-park” scene filmed outdoors. In the car-park scene, 6 of 258 trajectories represent the behaviour of actors. Both scenes are filmed using a single static camera.

The agents within each scene are tracked. In the case of the PETS2004 dataset, tracking data was provided alongside the videos. The agents within the car-park scene were tracked using a multi-purpose “blob-tracker” [7], set to output the position of object centroids in the image plane. Some post-processing was required to ensure a one-to-one mapping between objects in the scene and tracker output. Trajectories were then smoothed using a Kalman filter, and the velocity vector of the Kalman smoothing was stored for later use. This provides us with five measurements for each agent for each frame: x-position, y-position, time, and the x and y components of the velocity vector. These are the only aspects of the agents’ behaviour fed into the next stage of processing – all appearance information is discarded.

Within the car-park scene, the location and extent of any exits are learned from a training set of 200 pedestrian and car trajectories. As in [10] we find it convenient to learn a representation of the exits from the collection of entrance and exit points using a probability density expressed as a mixture of Gaussians. The models are trained using Cootes and Taylor’s Kernel version of the

¹so called because it is from the PETS2004 dataset, generated as part of the EC Funded CAVIAR project/IST 2001 37540

Expectation-Maximisation (EM) algorithm [2], initialised with K-means. Figure 1 (a) shows the exit model learned for the car-park scene. Due to insufficient data, this learning was not carried out with the PETS2004 dataset, and the exit model was created by hand.

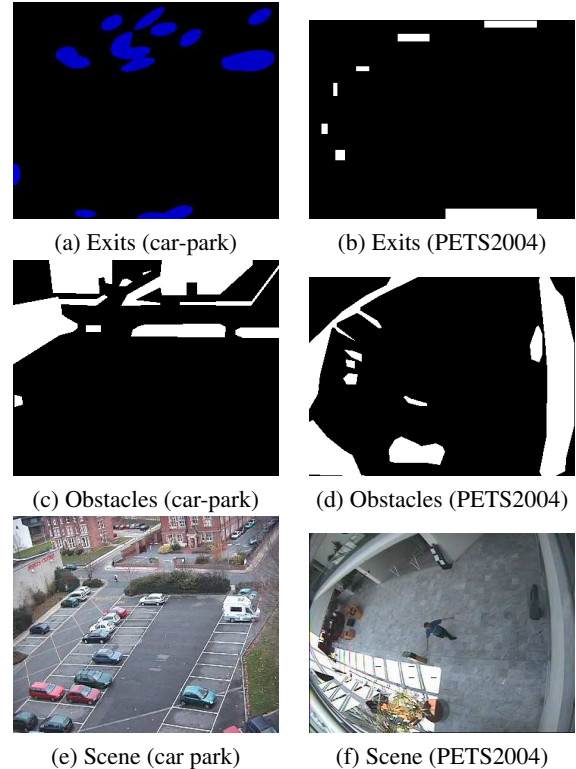


Figure 1: The exit model, obstacle model and scene for the car-park (left) and PETS2004 (right) scenes.

The obstacle model is first hand crafted, by taking a still image of the scene and marking those sections that agents would not be able to walk through or across (hedges, trees, buildings and so on). This bitmapped model was then converted into a polygonal representation by straight line approximation as detailed in [8]. As with the exit model and the tracks of the agent, the obstacle model is represented within the image plane. The obstacle models are shown in Figure 1 (c) & (d). This is an extension of the previous work [3, 4] in which the obstacle model was represented as a bitmap: using a polygonal representation escapes the problems which occur around the sawtoothed corners of bitmapped obstacles and also presents an improvement in computational speed.

2.1 Determining goal-directedness

In order to capture something of the way in which human agents navigate through a scene, we use the concept of a

sub-goal. We assume that people navigate through a scene in a piecewise linear fashion, turning at the vertices of obstacles. If, for example, there is one obstacle between an agent and their final goal, the agent first travels in a straight line towards a point on or near one of the vertices of the obstacle, then changes direction, then travels onwards towards their goal. These virtual, intermediate goals are what we call *sub-goals*, and they are the places within a scene where an agent might choose to change direction.

The algorithm we propose for measuring intentionality has two main phases:

- Firstly, we work out which goals within a scene might be targets of a particular agent for a particular frame, by assigning a “state” to each goal for each agent. The states represent whether the agent is headed towards the goal or away from it, or whether the agent is headed towards a sub-goal which might eventually lead to that goal.
- Secondly, we consider these goal categorisations over time and assign cost to those transitions associated with movement away from that goal. In this way, we end up with a cost associated with each goal in the scene and the most likely goal for the agent has the lowest cost.

To categorise the goals within the scene, we first label each open (i.e. non-obstacle) area of the scene. The polygonal obstacle model represents the obstacles as a list of ordered vertices \mathbf{v} , and we know the position \mathbf{x} and direction of travel θ of each agent. We can find the tangential vertices as follows: For each obstacle within the scene, consider each vertex \mathbf{v}_i in turn taking a line from \mathbf{x} through that vertex. If the neighbouring vertices (\mathbf{v}_{i+1} and \mathbf{v}_{i-1}) are both on the same side of the line through \mathbf{x} and \mathbf{v} , then \mathbf{v} is a tangential vertex on that obstacle. In order for a tangential vertex to be a potential sub-goal, it must be visible from \mathbf{x} , and the agent must be headed towards it: that is, the line from \mathbf{x} to \mathbf{v} must not pass through any other obstacles. Visible tangential vertices are considered sub-goals if the agent might be headed towards them - that is, the angle between \mathbf{x} and \mathbf{v} lies between $\theta - 1$ to $\theta + 1$.

Sub-sub-goals can be discovered in an analogous fashion simply by repeating the process with the location of the sub-goal in the place of \mathbf{x} , and the polygon representing the area already visible treated as another, virtual, obstacle. The polygon of already visible space is treated as an obstacle to prevent paths to sub-sub-goals crossing areas of the scene already visible. This procedure is then continued recursively until the entire scene is classified. This provides a significant improvement over the previous implementation[3, 4] which was capped at two levels of sub-goal analysis.

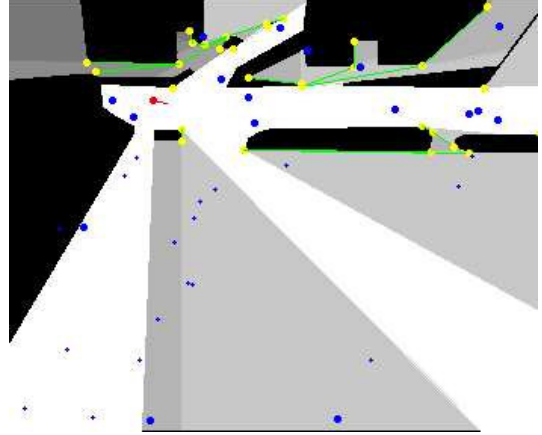


Figure 2: An example of the sub-goal algorithm in action. The agent is towards the left of the picture near the top, represented as a red dot with a line indicating direction of travel. The polygon of directly visible space is coloured white, and successive levels of sub-goal area are represented with successively darker shades of grey. Sub-goals themselves are yellow dots, and the green lines represent paths between sub-goals. Goals are represented in blue.

The output of this stage of classification is a completely labelled scene for each frame, which takes into account the current position and direction of motion of the agent: it is a form of agent-centric map. There are three types of relationship between an agent and each goal for each frame, which can be determined from the map at the position of the goal $Label(x_g)$, and the angle ϕ , which is the angle subtended by a line between the position of the goal \mathbf{x}_g , the position of the agent \mathbf{x} , and the agent’s current direction θ . These are

1. S_0, S_1, S_2, S_n : The goal is either directly visible and the agent is heading towards it $-1 < \phi < 1$; or the goal is accessible via a sub-goal (or two, or three...) Goals in one of these states are a potential explanation for the agents’ trajectory.
2. D : The goal is directly visible to the agent; but they are heading away from it: $\phi > 1$ or $\phi < -1$.
3. N : The goal is not visible to the agent (it is on the other side of an obstacle, and is not reachable by means of a sub-goal) .

2.2 Analysing patterns of goal activity over time

The following stage of analysis provides a unification of these frame-by-frame classifications in order to determine whether or not a particular goal is a viable *explanation* for

the trajectory as a whole. Essentially, we look at the pattern of state transitions associated with each goal in turn, asking the question “*Is this a possible explanation for the agent’s behaviour?*” or “*Could they be headed towards this goal?*”. With goals near the boundary between labels, noise in the direction measurement can cause noise in the categorisation. To minimise the effects of this noise, classification information is “smoothed” by voting over a five frame moving window: for each frame, the categorisation of each goal is replaced by the most common categorisation (the mode).

Those goals which are consistent or reasonable explanations for the behaviour so far will be those goals whose patterns of state transitions are consistent with motion towards that goal. In the simplest case, where an agent can see their goal from the outset and heads directly towards it, the goal would be in state S_0 from the start of the trajectory to the finish. If, however, an agent’s final goal is two levels of indirection away from his or her start position we can expect a pattern of transitions of the sort $S_2 \rightarrow S_1 \rightarrow S_0$, probably staying in any or all of these states for some number of frames. To create a measure of intentionality, we associate a cost with those state transitions associated with travel away from a particular goal. Thus, simple goal-directed behaviour of *moving-towards* is cost-free, and behaviour inconsistent with a particular goal implies a penalty. Costs are calculated for each goal within the scene using the state transition diagram shown in Figure 3, providing us with a measure of how good an explanation each goal is for the trajectory so far.

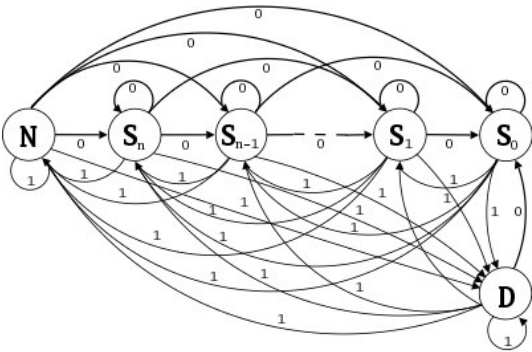


Figure 3: State transition diagram for calculation of cost

There are a number of possible uses for such scores within a surveillance situation. It would be possible, for example, to label particular goals or areas of scene as being “suspect”, and to filter the surveillance data on this basis. What we propose however is a more general approach. Taking the lowest cost goal for a trajectory as being the cost of its *best explanation*, we have a measure of the *intentionality* or *goal-directedness* of that trajectory. Finally, the cost is

then normalised by dividing by the number of frames thus far. This statistic provides a measure C where $1 \geq C \geq 0$.

3 Using intentionality in a surveillance context

In this section we consider the possible ways in which such a measure can be used for surveillance purposes, and ways in which we can evaluate its usefulness. The evaluative approach favoured here is one which exploits a result from [12], who found that naïve observers perform as well as security guards when it comes to anticipating unusual or criminal behaviour. Given this result, a group of naïve observers look at each trajectory in our surveillance footage, and rank them on a scale of 1 (uninteresting) to 5 (interesting). For each agent, a separate movie containing only those frames of video which encompass that agent’s trajectory was produced with the agent of interest clearly highlighted throughout. Volunteers were asked to rate each agent’s behaviour as detailed above. For the car-park dataset, 7 volunteers ($n_s = 7$) ranked the 258 trajectories ($n = 258$) and for the PETS dataset, 12 volunteers ($n_s = 12$) ranked 22 trajectories ($n = 22$). We propose using the mean rank as an indication of the overall *interestingness* of a trajectory; we can calculate correlations between the output of the software and the human ranks (see [4] for a much fuller treatment of this issue), and we can use the human results to set some threshold below which we decree that trajectories are uninteresting.

3.1 Correlation results

As this data is non-parametric and on different scales, there are two correlation statistics which are applicable. These are Spearman’s Rho and Kendall’s Tau [1]. Spearman’s Rho (r_s) is calculated by first ranking the data and then performing a Pearson’s product moment correlation calculation on the resultant ranks using Equation 1 in which x_i and y_i are matched pairs of ranks.

$$r_s = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{[\sum_{i=1}^n (x_i - \bar{x})^2][\sum_{i=1}^n (y_i - \bar{y})^2]}} \quad (1)$$

Kendall’s Tau (T_k) operates differently. Whilst it utilises the same underlying information (ranks of scores), it is not directly comparable to Spearman’s Rho. Instead of relying upon the numerical difference between ranks, it only takes account of the relative orderings of ranks. To calculate T_k , one must first work out the total number of concordant and discordant ranks. The formula for calculating T_k is given in Equation 2, in which T_x and T_y are the terms correcting for tied ranks.

	PETS2004 C		Car-park C	
	r_s	T_k	r_s	T_k
Mean Human	0.74	<i>0.56</i>	0.43	0.36

Table 1: Correlation statistics comparing the mean human rank with the C score

$$T_k = \frac{\text{concordant} - \text{discordant}}{\sqrt{n(n-1) - T_x} \sqrt{n(n-1) - T_y}} \quad (2)$$

The between-human correlation matrices show that there is a high level of agreement between the subjects. In the car-park dataset, all 21 T_k correlation coefficients were positive, and significant at the 0.0001 (0.1%) level, as were the 21 r_s measurements. In the PETS2004 dataset, using T_k , 57 of the 66 measurements were significant at the 0.001 (1%) level, and 44 of these were also significant at the higher 0.0001 (0.1%) level. r_s provided slightly less significant results with only 48 of the PETS2004 correlations being significant at 0.001 and 33 also at 0.0001.

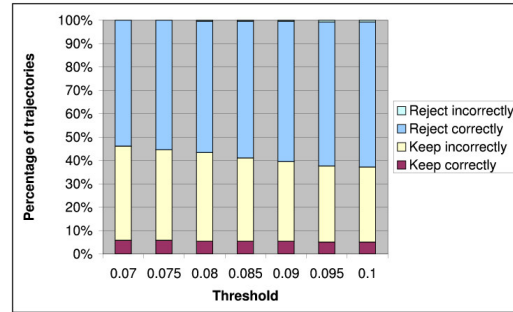
Correlations between the mean human ranking and the machine generated C score are shown in Table 1. Those results significant at the 0.0001 level are shown in boldface and those significant at the 0.001 in italics. All correlations are positive and significant.

3.2 Filtering

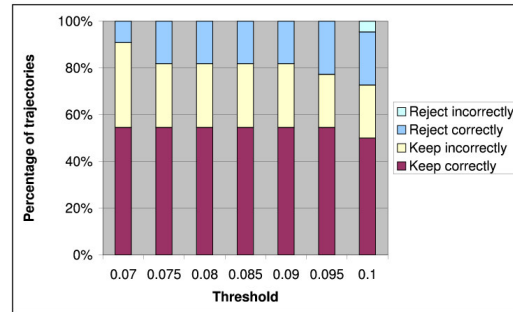
The application we are considering here involves simple thresholding on the C score to remove those scenes in which we are certain nothing of interest is happening. If we set a suitably low threshold upon the human ranks (H_t) below which we consider behaviour to be *uninteresting* we can then determine whether or not our C score could be used to automatically reject some proportion of these clearly dull behaviour patterns. There are two ways in which this can be done.

1. *By trajectory*: This is the simplest of the measures - we have trajectory-by-trajectory indications of both C and human opinion. This is less realistic than the second option as it fails to take into account situations where more than one person is in the scene.
2. *By frame*: This is a more complicated measure, as it involves converting by-trajectory measures of cost and human rank into by-frame measures by taking the highest scoring trajectory per frame as a measure of intentionality for that frame. However it is a more realistic approach, as within real surveillance situations filtering would need to be based upon whole scenes rather than individual trajectories.

The threshold chosen for H_t is intentionally very low. The aim is to provide a filter which will remove a proportion of completely uninteresting footage whilst leaving as much as possible of the interesting footage. The value of 2 was chosen after inspection of the videos and consideration of the comments made by the volunteer observers. This threshold provides us with 15 trajectories within the car-park dataset which are considered to be *interesting* and 243 which we would wish to filter out. Within the PETS2004 dataset the same threshold has 12 trajectories which we would wish to keep and 10 which we should ignore².



Car-park dataset



PETS2004 dataset

Figure 4: The effect of thresholding by trajectory

From the charts shown in Figure 4 it is clear that filtering on C scores as suggested would preserve a number of uninteresting trajectories as well as those which are considered interesting - it would not be a good deal of use as an interesting behaviour detector as the number of *false positives* is high. However, given our stated aim of creating a boring behaviour rejector the results are much more promising: it is important to reject as boring as many trajectories as possible whilst rejecting *no* interesting trajectories by mistake. From the charts, a threshold of around 0.09 seems to be the most effective, rejecting none of the interesting trajectories

²The large difference in proportion of interesting behaviours here is due to the slightly contrived nature of the PETS2004 dataset

within the PETS2004 dataset, and rejecting only one interesting trajectory from the car-park dataset (which turns out to be a person using an unusual shortcut).

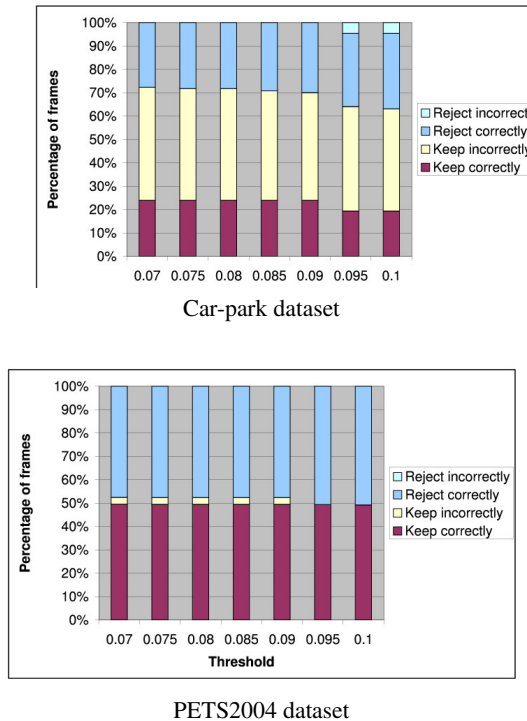


Figure 5: The effect of thresholding by frame

The results for thresholding on a frame-by-frame basis (shown in Figure 5) rather than upon entire trajectories show that a smaller proportion of the footage would be rejected. One possible factor in this is that those trajectories which are considered to be *interesting* tend to be longer, and hence tend to influence the C scores of more frames. This is obvious and matches our initial hypothesis well: agents who cross the scene purposefully heading directly to their goal do not take long to cross the field of view of the camera. The charts shown in Figure 5 support our earlier suggestion of a threshold on C of around 0.09.

4 Summary and Conclusions

This paper has shown that a cognitive vision system could usefully be used in a surveillance application to automatically ignore those trajectories which represent intentionally explicable behaviour. On a trajectory by trajectory basis, the system could be used to exclude from surveillance the actions of around 60% of individuals in a typical outdoors scene. Such a system has obvious applications in real world

surveillance, where hundreds of cameras exist but only a small fraction can be monitored. As it stands, the system is applicable to scenes in which people do not linger. However, the introduction of *inactivity zones* as in [10] could extend the applicability of the system to richer scenes, such as those containing ATMs or other areas where people gather.

References

- [1] Clarke G.M. and Cooke D. *Nonparametric systems for the behavioral sciences*. McGraw Hill, Singapore, 1988, 2 edition.
- [2] Cootes T. and Taylor C. ‘A mixture model for representing shape variation.’ In: *Proc. British Machine Vision Conference (BMVC)*, pp. 110–119. 1997.
- [3] Dee H.M. and Hogg D.C. ‘Detecting inexplicable behaviour.’ In: *Proc. British Machine Vision Conference (BMVC)*. Kingston-on-Thames, UK, 2004.
- [4] Dee H.M. and Hogg D.C. ‘Is it interesting? comparing human and machine judgements on the pets dataset.’ In: *ECCV-PETS: the Performance Evaluation of Tracking and Surveillance workshop at the European Conference on Computer Vision*. Prague, Czech Republic, 2004.
- [5] Jan T., Piccardi M. and Hintz T. ‘Detection of suspicious pedestrian behavior using modified probabilistic neural network.’ In: *Proc. of Image and Vision Computing*, pp. 237–241. Auckland, New Zealand, 2002.
- [6] Johnson N. and Hogg D.C. ‘Learning the distribution of object trajectories for event recognition.’ *Image and Vision Computing*, Vol 14(8), pp. 609–615, 1996.
- [7] Magee D.R. ‘Tracking multiple vehicles using foreground, background and shape models.’ *Image and Vision Computing*, Vol 22, pp. 143–155, 2004.
- [8] Magee D.R. and Boyle R.D. ‘Building shape models from image sequences using piecewise linear approximation.’ In: *Proc. British Machine Vision Conference (BMVC)*, pp. 398–408. 1998.
- [9] Makris D. and Ellis T. ‘Spatial and probabilistic modelling of pedestrian behaviour.’ In: *Proc. British Machine Vision Conference (BMVC)*, pp. 557–566. Cardiff, UK, 2002.
- [10] Nait Charif H. and McKenna S.J. ‘Activity summarisation and fall detection in a supportive home environment.’ In: *Proc. International Conference on Pattern Recognition (ICPR)*. Cambridge, UK, 2004.
- [11] Smith G.J.D. ‘Behind the screens: Examining constructions of deviance and informal practices among cctv control room operators in the uk.’ *Surveillance and Society*, Vol 2(2/3), pp. 376–395, 2004.
- [12] Troscianko T., Holmes A., Stillman J., Mirmehdi M., Wright D. and Wilson A. ‘What happens next? the predictability of natural behaviour viewed through cctv cameras.’ *Perception*, Vol 33(1), pp. 87–101, 2004.