Real Time Target Tracking with Pan Tilt Zoom Camera

Pankaj Kumar, Anthony Dick
School of Computer Science
The University of Adelaide
Adelaide, Australia
Email: pankaj.kumar@adelaide.edu.au,
anthony.dick@adelaide.edu.au

Tan Soo Sheng
Electrical and Computer Engineering
National University of Singapore
Singapore

***Abstract*—We present an approach for real-time tracking of a non-rigid target with a moving pan-tilt-zoom (PTZ) camera. The tracking of the object and control of the camera is handled by one computer in real time. The main contribution of the paper is method for target representation, localisation and detection, which takes into account both foreground and background properties, and is more discriminative than the common colour histogram based back-projection. A Bayesian hypothesis test is used to decide whether each pixel is occupied by the target or not. We show that this target representation is suitable for use with a Continuously Adaptive Mean Shift (CAMSHIFT) tracker. Experiments show that this leads to a tracking system that is efficient and accurate enough to guide a PTZ camera to follow a moving target in real time, despite the presence of background clutter and partial occlusion.**

***Keywords*-Target Representation; CAMSHIFT; Tracking; PTZ camera;**

I. Introduction

Video surveillance is becoming ubiquitous in many public places including airports, train stations, shopping malls, and car parks. Although surveillance cameras are typically monitored manually, it is becoming increasingly common for some of the more mundane tasks to be performed automatically by video analysis software. In this paper, we focus on the task of tracking a person as they move about in an environment. This underlies many higher level surveillance tasks including behaviour analysis [1] and intruder detection.

It is often the case that a target must be tracked across a larger area than is visible in a single camera's field of view. There are two possible solutions: one is that the target is tracked across cameras [2]–[4], and another is that the target is followed and tracked with a moving Pan-Tilt-Zoom (PTZ) camera [5], [6]. The later approach is sometimes more desirable than former because, firstly, in cross camera tracking the target may be lost in the blind regions between cameras, and secondly, it is very difficult (nearly impossible for real environments) to lay out the cameras in such a way that there is always overlap between the field of views (FOVs) of the cameras. In real world environments it is also difficult to calibrate the cameras. Following a target with a moving camera gives the operator (human or software) more flexibility to adjust the camera in response to events so that a clear view of the target of interest is obtained. Our work here focuses on tracking a target of interest with a moving PTZ camera.

Computer Vision algorithms which are part of a real time video surveillance system are expected to be computationally efficient. They must be able to track the target in real time, process the signals for camera control and still have computational resources for other services such as storing the video and processing user requests. Continuously Adaptive Mean Shift (CAMSHIFT) [7] has been shown to be accurate and reliable, but is primarily designed to track a target based on a static colour distribution, such as skin colour, which has been built offline and is not updated while tracking. There has been different methods for improving CAMSHIFT tracking method [8] [9]. In [8] Allen *et al.* in order to compute the probability of a pixel to belong to the target model, used a weighted multidimensional histogram, where the weight for a pixel's contribution to the histogram was computed using a simple monotonically decreasing kernel profile. In [9] spatial constraint of face and upper body and hence joint colour-spatial distribution was used to improve over traditional CAMSHIFT tracking. We augment CAMSHIFT to moving cameras by incorporating a novel way to obtain the target probability image. This is done by using an improved model of the target and a model of its background, and using Bayes theorem to compute the pixel weights for pixels which belong to the target.

A. Tracking System Overview

Our complete moving PTZ camera tracking system can be divided into three basic modules; (1) Initialisation, (2) tracking, and (3) camera control modules. Figure 1 gives the schematic block diagram of our tracking system.

The initialisation module is active when the PTZ camera is under manual control and monitoring the environment in which the user will identify the target to be tracked. Whenever the camera is stationary, a background model of the scene is built and masks of foreground moving targets are detected by background subtraction [10]. The user identifies a target for tracking by a mouse click on the target. The tracking can also be automatically initialise to track an object

978-0-7695-3866-2/09 $26.00 © 2009 IEEE
DOI 10.1109/DICTA.2009.84

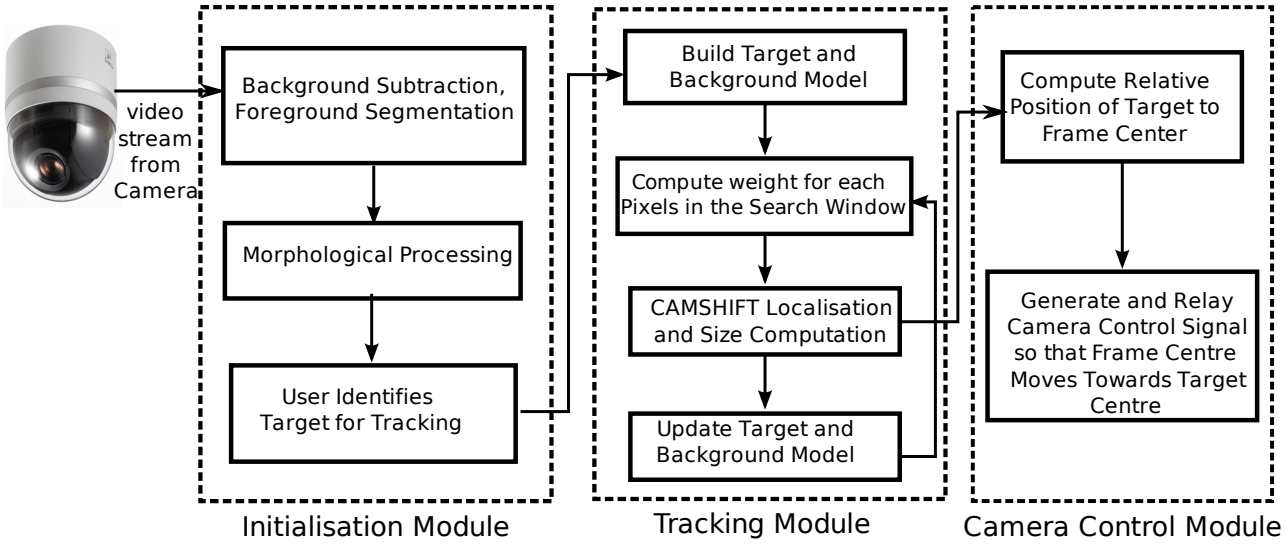


Figure 1. This figure shows the different modules of our complete tracking system, the flow of information, and processing of data.

which appear in a particular region of the camera's field of view. Even when the initialisation is automatic still the tracking with moving camera will run in real time.

Once a target has been identified, the tracking module takes up the task of tracking it. First, colour histogram models of the target and the surrounding background region are built using the current image and the foreground mask.

These histograms are then used to calculate a value for each pixel which defines the probability of its belonging to the target, relative to the probability that it belongs to the background.

This process is explained in detail in Section II. The image formed by these values is used as input to a CAMSHIFT filter, which iteratively finds the centre of mass of the target and its estimated size in the current frame. This is then used to control the camera, as described in Section IV.

When a new frame is captured, control passes back to the tracking module, with the updated estimate of target position and scale obtained from CAMSHIFT. The process of tracking and camera control is then repeated for the new image. Experimental results are shown in Section V.

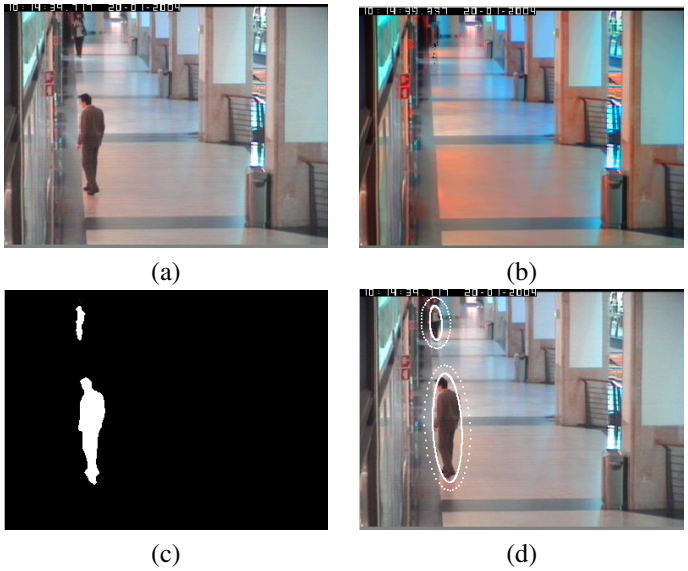


Figure 2. These images show the process of building the model of a target. (a) a frame from the image sequence (b) background model of the image sequence (c) foreground segmentation result (d) bounding ellipses. The target model includes a histogram of the target pixels and histogram of the pixels which is the region between solid and dotted ellipses.

II. Target Representation

Once a target has been manually identified by an operator or automatically, the first task of the tracking module is to construct a model that represents it. In our system, a target is represented by a histogram of the region which is detected as foreground and a histogram of a neighbouring region which is background. Figure 2 shows an example. The pixels inside the solid ellipse which are classified as foreground are used to build the foreground colour histogram T_{fg}, and pixels between the solid ellipse and dotted ellipse which are classified as background are used for building the background colour histogram T_{bg} for the target. The ellipse is the best fit for the foreground pixels, and is obtained by the method used in [11]. The background region between the solid ellipse and dotted ellipse is of fixed width of 15 pixels. We empirically came up with the width of 15 pixels. Alternatively, this width can be the maximum distance of the foreground pixel from the nearest boundary, computed by distance transform on the foreground blob [12].

We assume that for at least a short time period ($> 1s$) before the beginning of the tracking process, the PTZ camera is stationary. In practice this is not a heavy constraint, as

the user is unlikely to select a target while the camera is moving. Thus we can obtain a background model of the scene Figure 2(b) and a segmentation of the moving target as shown in Figure 2(c). Figure 2(d) shows the bounding ellipses for foreground region and background region.

A target's foreground model T_{fg} is a N-bin histogram, which is non-parametric colour probability distribution function (pdf) of the target foreground. Similarly the target's background model is another N-bin histogram.

$$T_{fg} = \{q_u^{fg}\}_{u=1}^{N}, \quad where \quad \sum_{u=1}^{N} q_u^{fg} = 1$$
$$T_{bg} = \{q_u^{bg}\}_{u=1}^{N}, \quad where \quad \sum_{u=1}^{N} q_u^{bg} = 1 \tag{1}$$

Let the function which associates bin index u to the colour vector at pixel location $x_i, \quad i = 1...n_{fg}$ where n_{fg} is the total number of pixels in the target foreground, be denoted by $b(x_i) \in \{1, ..., N\}$. The probability of the colour feature $u = \{1, ..., N\}$ in the target foreground is then computed as

$$q_u^{fg} = C_{fg} \sum_{i=1}^{n_{fg}} w_i^{fg} \delta[b(x_i), u]. \tag{2}$$

Here, δ is the kronecker delta function which is 1 when $b(x_i) = u$ and zero otherwise, and C_{fg} is the normalising constant such that $\sum_{u=1}^{N} q_u^{fg} = 1$. The term w_i is a weight for each pixel x_i, which depends on its position relative to the foreground mask. It is computed by a distance transform on the foreground blob as described in [12]. The effect is to give less weight to pixels that are near the edge of the foreground region, as they are considered to be less reliable. Figure 3 shows a pictorial representation of the weights of the pixels for foreground and background pixels. Figure 3(a) is the foreground mask obtained by using background subtraction algorithm as described in [10], 3(b) is an image of the weights for foreground pixels, 3(c) is an image of weights for the background pixels. The intensity of the pixels in images 3(b) and 3(c) is function of their weights. The higher the weight, the brighter the pixels are.

The probability of the colour feature $u = 1...N$ in the target background is computed as

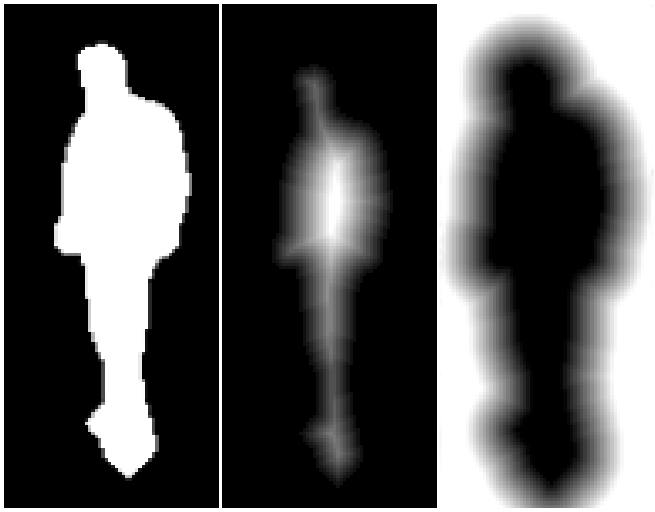
$$q_u^{bg} = C_{bg} \sum_{i=1}^{n_{bg}} w_i^{bg} \delta[b(x_i), u]. \tag{3}$$

where n_{bg} are the total number of pixels in the background region, and C_{bg} is a normalising constant.

A. Calculating the weight image

The histograms computed above are non-parametric conditional densities which can be denoted as $q_u^{fg} = P(u|fg)$ and $q_u^{bg} = P(u|bg)$. Using Bayes' formula the conditional distributions $P(fg|u)$ and $P(bg|u)$ can be computed as

$$P(fg|u) = \frac{P(u|fg)P(fg)}{P(u)}$$



(a) (b) (c)

Figure 3. The images here shows the weights of the foreground and background pixels used in building the target model. (a) is the segmentation result (b) is a plot of the weights for the foreground pixels (c) is a plot of the weight of the background pixels. Brighter the pixels are more is their weight in building the foreground and background histograms.

$$P(bg|u) = \frac{P(u|bg)P(bg)}{P(u)} \tag{4}$$

We use the ratio of $P(fg|u)$ and $P(bg|u)$ to compute a weight image of the target in the next frame, where each pixel's value is the relative probability of its membership in the foreground and background region:

$$\Omega_i = \frac{P(fg|b(x_i))}{P(bg|b(x_i))} = \frac{P(b(x_i)|fg)P(fg)}{P(b(x_i)|bg)P(bg)} \tag{5}$$

where $P(fg) = \frac{n_{fg}}{n_{fg}+n_{bg}}$ and $P(bg) = \frac{n_{bg}}{n_{fg}+n_{bg}} = 1 - P(fg)$. We next show how this image leads to improved target detection and tracking.

III. CAMSHIFT TRACKING

In its original form, CAMSHIFT uses a colour histogram representing target appearance to evaluate the probability that each pixel in a region belongs to foreground. It then iteratively shifts the region to converge on the local optimium: the region whose pixels have the highest combined probability of being foreground.

We improve on this by replacing the colour histogram back-projection with Equation 5 when calculating the probability for each pixel to belong to the target. Figure 4 compares the weight image computed by the method presented here and the weight image computed by the colour histogram. The weight image computed by our method is more accurate representation of the target than the previous approach.

The modified CAMSHIFT tracking algorithm can now be summarised by the following steps

(a) (b) (c)

Figure 4. The images here shows (a) the frame in which the weight image for the target closure to the camera is computed (b) colour probability image computed using histogram back-projection as used in [7] (c) the weight image computed by the method proposed here. Pixel intensity is proportional to probability of belonging to foreground. The weight image (c) is a more accurate representation of the target in frame (a) than the colour probability image (b).

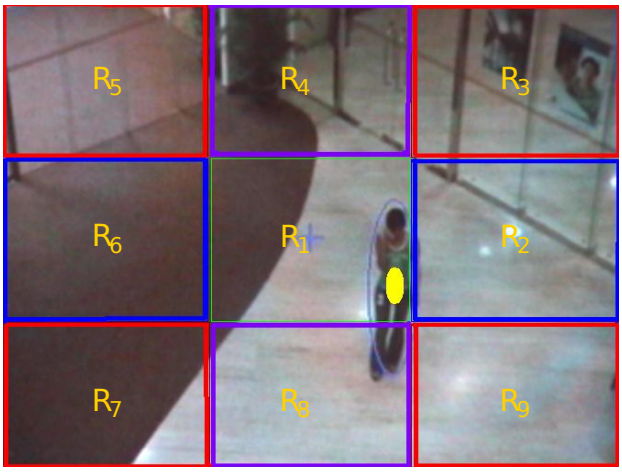


Figure 5. This figure illustrates the working of the camera control module. The aim of the camera control module is to keep the target centre in the region $\mathbf{R_1}$. The target centre by localisation can be found in any of the regions ranging from $\mathbf{R_1}...\mathbf{R_9}$. According to the region in which the target centre is found Pan or tilt or both commands are generated and send to the camera depending upon the last sent command.

1) Initial location of the 2D mean shift search window is based on the bounding box of the foreground-background segmentation blob of the target.
2) Compute the weight image of the 2D region centred at the search window location and the surrounding area, using Equation 5.
3) Use mean-shift to converge to the new location of the target. Store the zeroth moment and mean location.
4) Threshold the weight image to obtain a binary foreground/background mask for the target. Use this to update the foreground and background colour histograms. Note that this works whether the camera is static or moving.
5) In the next image frame the search window is centred at the new target location, and its size is the function of zeroth moment. The process is again repeated starting from step 2.

The foreground and background histograms are updated using a learning factor β_{fg} and β_{bg}, respectively. The value of β_{fg} and β_{bg} is empirically chosen and it depends upon the nature of lighting in the environment and environment itself, which determines the rate at which the model of the foreground and background will change. Usually β_{bg} is greater than β_{fg}. For a moving camera background model will change faster than foreground model of the target.

$$\begin{aligned} T_{fg}^{k} &= (1-\beta_{fg})T_{fg}^{k-1} + \beta_{fg} \times T_{fg}^{localised} \\ T_{bg}^{k} &= (1-\beta_{bg})T_{bg}^{k-1} + \beta_{bg} \times T_{bg}^{localised} \end{aligned} \tag{6}$$

where k is the current frame and $T_{fg}^{localised}$, $T_{bg}^{localised}$ are the foreground and background models for the target localised in the current frame.

IV. Camera Control

The aim of the camera control unit is to generate and relay control commands to the camera such that the target centre lies in the region "$\mathbf{R_1}$" of the frame as shown in Figure 5.

In every frame, the horizontal and vertical distance of the target from the frame centre in region "$\mathbf{R_1}$" is calculated. If the horizontal or vertical distances or both are greater than a threshold then commands are generated by giving priority to the distance which is greater. If vertical distance is greater than horizontal then tilt command is given priority over pan and vice-versa. If the target centre lies in "$\mathbf{R_1}$" region ie the horizontal and vertical distances are within threshold then stop command is send to the camera.

One concern in the design of a camera control procedure is to minimise the delay of the camera in executing commands. Different PTZ cameras have different amounts of lag; the camera which we used had a delay of approximately 100 ms in executing commands. To reduce the loss of time in transmission and execution of commands, the program keeps a record of the command the camera is executing. Hence, repeated commands will be avoided to reduce delays. For example if the target centre is in region "$\mathbf{R_2}$" then command for right panning is generated and checked against the last command send to the camera. If the last command is different from pan-right then first a stop command is send and after a delay of 100 ms the pan-right command is send. But if the last command is same as pan-right then no command is send to the camera which saves 200 ms of time. This is important from a tracking point of view as the speed of the target which the camera can track depends more on this factor, than on the computation required for tracking.

V. Results

We show three sets of tracking results of the several others, were the proposed PTZ camera tracker has been able to successfully track the identified individual with the moving camera. These sequences have been captured using a screen capture software while running the tracking implementation on the same computer. This is worth mentioning to show the computational efficiency of the proposed algorithm. The frame size of the frames processed by the tracking algorithm is 352×255 and all the 25 frames in a second are processed without 100% use of the CPU on a 3.0 G-Hz Pentium machine.

Due to the latency of the camera in executing instructions, in tracking results the camera centre lags behind the target centre when it is moving quickly. However, it recovers when the target slows down or stops. This is due to the slow processing of the commands by the PTZ camera and not due to the tracking algorithm.

In the results shown in Figure 6 there are instances of illumination changes, highlights and change of target pose. This tracking image sequence is 2980 frames long. The tracker has been successfully able to handle changes in the target and background. Some times the camera centre is not exactly on the target centre and lags behind the target centre due to the latency of the camera in executing the commands sent to it. This problem can be avoided by using a PTZ camera whose response to commands are faster.

Figure 7 shows tracking results for another person in the same scenario as Figure 6. In this case there are multiple moving targets which can potentially distract the tracker. As seen in Figure 7 (e) (f) the representation of the target is robust enough to continue tracking even in presence of these distractions. The length of this tracking sequence is 2814 frames.

Figure 8 shows successful tracking results in presence of partial occlusion. The target in a lecture theatre environment has been tracked in spite of being partially occluded by the chairs.

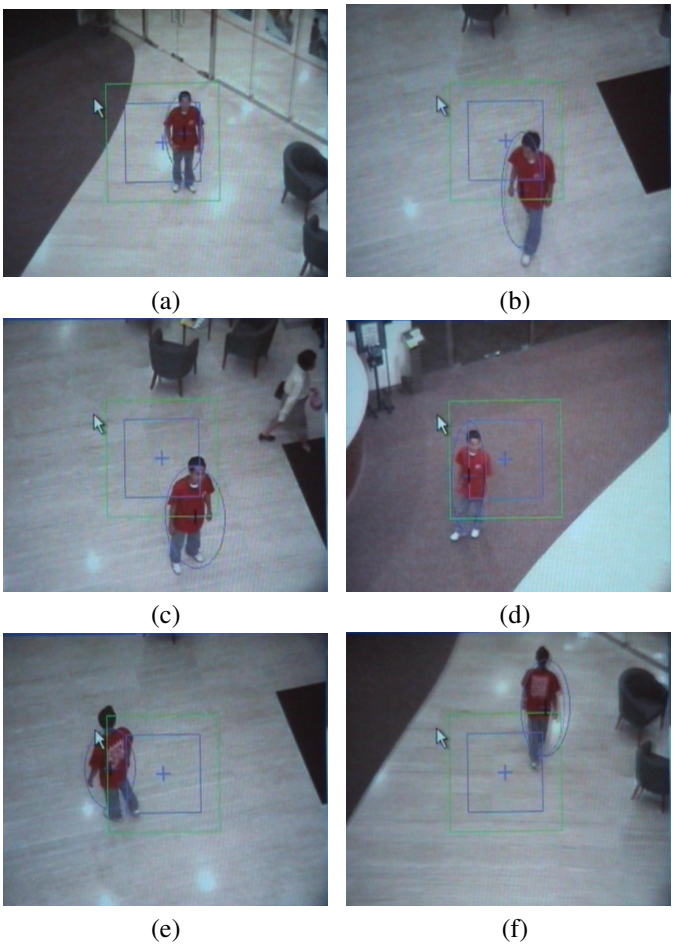


Figure 6. Successful tracking of a person with moving PTZ camera. There are instances of illumination change and background change. The complete tracking video can be seen at http://www.youtube.com/watch?v=QzMcM1Sn6cc (dicta_re1).

VI. Summary and Ideas for Future Work

In this paper a robust target representation and detection method has been proposed, based on distance transform of the segmented target and application of Bayes theorem to the probability distribution of the foreground pixels and background pixels in the neighbourhood of the target. Several tracking videos demonstrate the efficacy of the novel target representation in tracking a target with a moving PTZ camera. The target representation and hence localisation is better than the previously used histogram back-projection method. The representation and detection is also robust to distractions and partial occlusions. Some times the target representation is vulnerable to large scale illumination changes, which can be improved by a using colour space which is more robust to illumination changes. The vulnerability to illumination change can also be ameliorated by using multiple cue in detection of the target. It would be interesting to extend this method to multiple target tracking, which adds the extra problem of deciding who the camera will follow in case of clashes. Use of particle filter instead of CAMSHIFT filter which is more robust in tracking multiple targets can also be explored.

References

[1] P. Kumar, S. Ranganath, W. Huang, and K. Sengupta, "Framework for real time behavior interpretation from traffic video," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 1, pp. 43–53, 2005.

[2] 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 *Proceedings of British Machine Vision Conference*, 2004, pp. 397–406.

[3] O. Javed, Z. Rasheed, K. Shafique, and M. Shah, "Tracking across multiple cameras with disjoint views," in *Proceedings of Ninth IEEE International Conference on Computer Vision*, vol. 2, 2003, pp. 952–957.

Figure 7. Tracking results for another person in presence of colour distraction as another person passes the person being tracked. The complete tracking video can be seen at http://www.youtube.com/watch?v=QzMcM1Sn6cc (dicta_re2).

Figure 8. Tracking results for a person who has been partially occluded by the chairs. The complete tracking video can be seen at http://www.youtube.com/watch?v=QzMcM1Sn6cc (dicta_re3).

[4] L. Lee, R. Romano, and G. Stein, "Monitoring activities from multiple video streams: establishing a common coordinate frame," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 758–767, 2000.

[5] A. A. Argyros and M. I. A. Lourakis, "Tracking multiple colored blobs with a moving camera," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, 2005, p. 1178.

[6] S. Kang, J. Paik, A. Kosehan, B. Abidi, and M. A. Abidi, "Real-time video tracking using ptz cameras," in *Proceedings of the SPIE 6th International conference on Quality Control by Artificial Vision*, vol. 5132, 2003, pp. 103–111.

[7] G. R. Bradski, "Computer vision face tracking as a component of a perceptual user interface," in *In Proc. of the IEEE Workshop on Applications of Computer Vision*, 1998, pp. 214–219.

[8] J. G. Allen, R. Y. D. Xu, and J. S. Jin, "Object tracking using camshift algorithm and multiple quantized feature spaces," in *Proceedings of the Pan-Sydney area workshop on Visual information processing*. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2004, pp. 3–7.

[9] B. Kwolek, "Camshift-based tracking in joint color-spatial spaces," in *Lecture Notes in Computer Science : Computer Analysis of Images and Patterns*, 2005, pp. 693–700.

[10] P. Kumar, S. Ranganath, and W. Huang, "Queue based fast background modelling and fast hysteresis thresholding for better foreground segmentation," in *The Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing*, vol. 2, 2003, pp. 743– 747.

[11] P. Kumar, S. Ranganath, K. Sengupta, and W. Huang, "Co-operative multitarget tracking with efficient split and merge handling," *IEEE Transactions on Circuts and Systems for Video Technology*, vol. 16, no. 12, pp. 1477–1490, December 2006.

[12] P. Kumar, M. J. Brooks, and A. Dick, "Adaptive multiple object tracking using colour and segmentation cues," in *Asian conference on Computer Vision*, vol. 1, 2007, pp. 853– 863.