3D Surveillance System Using Multiple Cameras

Ajay Kumar Mishra, Bingbing Ni, Stefan Winkler, Ashraf Kassim

Department of Electrical and Computer Engineering
National University of Singapore
Singapore 117576

ABSTRACT

We propose a 3D surveillance system using multiple cameras surrounding the scene. Our application is concerned with identifying humans in the scene and then identifying their postures. Such information can help with automatic threat assessment of a scene. The cameras are fully calibrated and assumed to remain fixed in their positions. Object detection and interpretation are performed completely in 3D space. Using depth information, persons can easily be separated from the background and their posture identified by matching with 3D model templates.

1. INTRODUCTION

Current camera surveillance systems are essentially video recording systems used for post-event analysis. These systems are without any intelligence and hence are not able to provide sufficient security for the area under surveillance. Typically, there are security personnel watching the videos to check for any unusual activities. For obvious reasons of inconsistent human attention, such monitoring is very unreliable. Therefore, it is important to incorporate intelligence into video surveillance such that the system can raise alarms on the occurrence of any suspicious events.

Surveillance typically involves three primary tasks: foreground detection, object identification, and tracking. Until now, most of the surveillance work has been done in a 2D setup. A comprehensive summary of 2D surveillance systems can be found in the paper by Ismail et al.[1] In most existing systems, a single camera is used to observe an area. Foreground detection is generally done using background subtraction.[1] It involves subtracting the background image from the current frame; the pixels with a difference higher than a given threshold constitute the foreground. Due to possible changes of the background, temporal information[2] is used to dynamically update the background image as well. However, the performance of background subtraction methods is still largely restricted to cases where lighting does not change a lot and where things that move are the objects of interest. Although there are some statistical techniques which can adapt to changing lighting conditions as well as the appearance and disappearance of objects from the background, the limitation of using 2D information alone reduces the effectiveness of such a system. Once the foreground is detected, the object is generally identified using template matching techniques. In this paper, we are mainly concerned with the first two tasks, namely foreground detection and object identification, which are prerequisites for the problem of tracking.

Using 2D techniques it is very hard to resolve occlusions and track multiple people, whereas 3D techniques can take care of the occlusion problem and can distinguish multiple objects from the scene quite accurately. A few attempts have been made to use 3D data in surveillance, but they are mostly restricted to finding the location of the detected object in the image, and partially resolving occlusions. There has been some work done using stereo cameras,[3–5] where a camera pair is used to obtain a depth map of the scene, which then becomes the basis for extracting human actors from the background. These methods still face the problem of occlusion. Occlusion handling has always been a major concern for visual surveillance of natural scenes with many moving objects and actors. Using multiple cameras around the scene can help alleviate the occlusion problem. Closest to our approach is the work by Müller et al.[6] They reconstruct the 3D scene with the objective to assist post event analysis of the scene, but do not attempt to use the 3D information to achieve better object identification and occlusion handling.

E-mail of corresponding author: ajay.mishra@nus.edu.sg

We use multiple stereo camera pairs surrounding the scene to create a 3D reconstruction of the scene. The cameras are fully calibrated, meaning their intrinsic as well as extrinsic parameters are determined a priori. After performing correspondence matching between two views, the different stereo pairs generate 3D points in the scene. These points are clustered into background and foreground. The foreground clusters are then compared with a 3D human model to detect the presence and posture of a human in the scene.

The paper is organized as follows: Section 2 describes our setup and calibration procedure. Section 3 illustrates our 3D reconstruction method with a detailed analysis of camera arrangement and epipolar geometry used. Section 4 explains 3D shape matching. Section 5 highlights some of our experimental results. Finally, section 6 concludes the paper and presents the scope for future work.

2. SETUP AND CALIBRATION

2.1. Setup

Our camera arrangement is shown in Fig. 1. The cameras are arranged in two pairs, each of which is placed at almost equal distances surrounding the scene. Each pair of cameras is used as a short-base stereo pair, which gives good depth information. The wide-line arrangement between the pairs of cameras helps resolve occlusions between objects in the scene, assuming the occlusion is absent in at least one of the short-base stereo views.

Figure 1. Experimental setup with two pairs of cameras separated by 90 degrees.

2.2. Calibration

In order to do metric reconstruction, we need to calibrate the cameras. We do so using ground truth. Since our cameras are going to be stationary we only need to calibrate them once. Although there are many different methods of calibration, we choose the direct linear transformation method[7,8] due to its simplicity. We also assume a-priori knowledge of certain 3D World coordinates $\mathbf{X}_i = (x_i, y_i, z_i)^T$ in the scene together with the corresponding 2D image coordinates $\mathbf{u}_i = (u_{xi}, u_{yi})^T$, for which the correspondence is determined manually. These points are converted into homogeneous form $\widehat{\mathbf{u}_i} = (u_{xi}, u_{yi}, 1)^T$ and $\widehat{\mathbf{X}_i} = (x_i, y_i, z_i, 1)^T$. The camera projection matrix P maps $\widehat{\mathbf{X}_i}$ into $\widehat{\mathbf{u}_i}$:

$$\widehat{\mathbf{u}_i} = P\widehat{\mathbf{X}_i},$$

$$\begin{bmatrix} u_{xi} \\ u_{yi} \\ 1 \end{bmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix}. \tag{1}$$

Substituting for 1 from the last row in this equation yields:

$$\begin{aligned} u_{xi}\,(p_{31}x_i + p_{32}y_i + p_{33}z_i + p_{34}) &= p_{11}x_i + p_{12}y_i + p_{13}z_i + p_{14} \\ u_{yi}\,(p_{31}x_i + p_{32}y_i + p_{33}z_i + p_{34}) &= p_{21}x_i + p_{22}y_i + p_{23}z_i + p_{24} \end{aligned}$$

$$\begin{pmatrix} x_i & y_i & z_i & 1 & 0 & 0 & 0 & 0 & -u_{xi}x_i & -u_{xi}y_i & -u_{xi}z_i & -u_{xi} \\ 0 & 0 & 0 & 0 & x_i & y_i & z_i & 1 & -u_{yi}x_i & -u_{yi}y_i & -u_{yi}z_i & -u_{yi} \end{pmatrix} \mathbf{p} = \mathbf{0}. \tag{2}$$

$\mathbf{p} = (p_{11}, p_{12}, p_{13}, p_{14}, p_{21}, p_{22}, p_{23}, p_{24}, p_{31}, p_{32}, p_{33}, p_{34})^T$ is a vector containing rows of the calibration matrix put together. Each known 3D point along with its corresponding 2D image coordinates gives two equations. As $\mathbf{p}$ is a 12-dimensional vector, we need at least 6 known 3D point correspondences. In a case where the number of known 3D scene points $N > 6$, the problem is overconstrained, and a least squares method can be used to obtain the best solution to $\mathbf{p}$. Finally, we obtain the calibration matrix $\mathbf{P}$ by rearranging the elements of $\mathbf{p}$.

3. 3D RECONSTRUCTION

As explained earlier, we reconstruct the scene using stereo views obtained from two different angles. The results are combined to generate the complete 3D information of the scene. There are various local and global methods to do dense stereo reconstruction. Global methods are computationally very expensive, which is why we use a local stereo reconstruction method based on the normalized sum of squared difference ($NSSD$) criterion.[9] The purpose of our system is to find people in the scene for tracking and not to generate a photorealistic 3D representation. Consequently, dense depth information can be sacrificed for more accurate but sparse 3D information from feature-based stereo.

3.1. Correspondence Matching

To obtain the 3D coordinate of a point, we need to find the correspondence between image pixels in the left and right image. Correspondence matching[10] is done by comparing a window patch extracted from around the reference point in the left image and comparing it with a patch extracted from around the pixel location in the right image. The comparison itself is done using $NSSD$:[9]

$$NSSD = \sum_{u,v} \left(\frac{(I_1(u,v) - \overline{I}_1)}{\sqrt{\sum_{u,v}(I_1(u,v) - \overline{I}_1)^2}} - \frac{(I_2(u,v) - \overline{I}_2)}{\sqrt{\sum_{u,v}(I_2(u,v) - \overline{I}_2)^2}} \right)^2, \tag{3}$$

where $I_1(u,v), I_2(u,v), \overline{I}_1, \overline{I}_2$ are patches from the left and right images and their mean values, respectively. (u,v) is a pixel coordinate inside the patch. The choice of patch size can affect the efficiency of the match and should thus be based on the texture content of the image. For images that are densely textured, the window size can be reduced compared to lightly textured images. Here we selected a square patch of 20×20 pixels in size. The pixel location corresponding to the minimum $NSSD$ value represents the most suitable match.

3.2. Epipolar Geometry

Finding correspondence between features in two views is central to the method of obtaining depth information about the point. In order to look for a match, a search is performed at all pixel locations in the image. This is computationally very expensive. Using epipolar geometry, we can restrict the search to one dimension only, making the matching process more efficient. In essence, we can project 3D points onto the 2D camera image with the help of the camera calibration matrix P. Using the pseudo inverse of the calibration matrix P^+, each image coordinate $\mathbf{x}$ can be backprojected into a ray passing through the camera center $\mathbf{C}$. The equation of this ray is:

$$\mathbf{X}(\lambda) = P^+\mathbf{x} + \lambda\mathbf{C}, \tag{4}$$

where λ corresponds to a given point on the ray.

Consider a pair of stereo cameras with P and P' as the camera projection matrices for left and right view, respectively. $\mathbf{X}(\lambda)$ is the equation of the ray passing through the left camera center and a given point in the left image. The ray can be projected onto the right view by multiplying it with the second camera matrix P' giving $\mathbf{x}_2(\lambda) = P'\mathbf{X}(\lambda)$. As λ ranges from 0 to ∞, a lot of projected image points $\mathbf{x}_r(\lambda)$ would lie outside the boundary of the image. Hence, it is necessary to restrict the range of λ such that $\mathbf{x}_r(\lambda)$ lie inside the image.

If the image is of size $m \times n$, the four corners of the image are $(0,0,1)^T$, $(0,m,1)^T$, $(n,m,1)^T$ and $(n,0,1)^T$. Using Eqn. (4), the points are backprojected to form rays $\mathbf{X}(\lambda)$; then choosing $\lambda = 1$ for each ray gives us homogeneous coordinates $\mathbf{X}_{ul}, \mathbf{X}_{ll}, \mathbf{X}_{lr}, \mathbf{X}_{ur}$. Now, a consecutive pair of 3D points together with the camera center $\mathbf{C}_r$ form a plane in 3D space. For instance, the equation of the plane passing through $\mathbf{X}_{ul}, \mathbf{X}_{ll}$ and $\mathbf{C}_r$ is given by its normal vector $\mathbf{n}$ such that

$$\begin{bmatrix} \mathbf{X}_{ul}^T \\ \mathbf{X}_{ll}^T \\ \mathbf{C}_r^T \end{bmatrix} \mathbf{n} = 0. \tag{5}$$

λ_l corresponding to the point of intersection between the ray $\mathbf{X}(\lambda)$ and the plane $\mathbf{n}$ satisfies

$$\langle \mathbf{n}, \mathbf{X}(\lambda_l) \rangle = 0.$$

Similarly, we can calculate $\lambda_r, \lambda_t, \lambda_b$ corresponding to the intersection of the ray with planes passing through $\mathbf{C}_r$, and other consecutive corners forming the right, top and bottom edges of the image, respectively. As a result, we can limit the range of λ from $\min(\lambda_l, \lambda_r, \lambda_t, \lambda_b)$ to $\max(\lambda_l, \lambda_t, \lambda_b, \lambda_t)$.

We calculate the edge map for the left and right image and then, for each edge point in the left edge map, the best match is determined in the right image using correspondence matching along with epipolar geometry. The best matching pixel corresponds to λ'. When $\mathbf{X}(\lambda')$ is substituted back into ray equation $\mathbf{X}(\lambda)$, we obtain the 3D coordinates $\mathbf{X}(\lambda')$ of the pixel in the left image.

4. 3D SHAPE MATCHING

Shape descriptors, i.e. 2D contours, 3D surfaces and 3D volumes, have been intensively studied in various application fields such as object recognition, shape database retrieval and posture recognition in terms of shape representation and shape similarity measurement. But these approaches are not quite suitable for the representation of deformable shapes such as the cluster of 3D points of a person. Recently, a novel shape descriptor known as *shape context*[11–13] has shown flexibility and robustness in the aspects of representing deformable shapes and measuring shape similarity. Since it is a global statistical descriptor, shape context preserves the localization of geometric features. It also has the useful properties of invariance to rotation, translation, scaling and is insensitive to small local shape variations. In this section, we present a modified version of the generic shape context descriptor to achieve our goal of human posture recognition.

4.1. 3D Shape Context Computation

The generic 3D shape context computation is performed by defining a reference frame containing the points cluster of the observed object. The selection of the type of reference frame should be tailored to the problem at hand. In our application, we use a cylinder as reference frame, which is flexible and compact based on the assumption that the human remains in a standing posture. The 3D shape context computation procedure includes reference point selection, normalization and shape distribution computation, which will be presented below.

4.1.1. Reference Point Selection

In a generic shape context computation, each point will be selected once as a reference point, and the shape distribution histogram will be computed from the 3D vectors generated from this reference point to other remaining points. Given N points and assuming that every shape distribution histogram is d-dimensional (d being the number of histogram bins) the full shape context descriptor is $N \times d$ in length, which is too large to be impractical and needs to be reduced. In our case, since the object of interest in the scene is assumed to be a standing person, we can select only one "robust" point as reference and compute the shape distribution with respect to it, which results in a d-length shape descriptor. Practically, we choose this reference point $\mathbf{R}$ as the center of the projection of all points to the ground, which is the $z = 0$ plane. Intuitively, the selected point is invariant to translation and rotation along the principal axis of the cylinder reference frame. Since human posture is not always symmetric, we choose M median values and then calculate their mean. These points are more likely to be the projection of the torso region. A typical value for M is $0.6N$. We found this to be robust even in case of noisy data points.

4.1.2. Normalization

Since we have a well-defined reference point $\mathbf{R}$ together with the reference frame, the normalization procedure is quite straightforward. Given a set of N 3D object surface points $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2 ... \mathbf{X}_N\}$, and the reference point $\mathbf{R}$ in Cartesian coordinate system, we first generate a set of vectors from $\mathbf{R}$ to $\mathbf{X}$, which is denoted as: $\mathbf{Q}_n = \mathbf{X}_n - \mathbf{R}$, where each $\mathbf{Q}_n$ in $\mathbf{Q}$ is a 3-vector (q_x, q_y, q_z) in Cartesian coordinates. Further we transform $\mathbf{Q}_n$ into the cylinder reference frame centered at $(0, 0, 0)$. The transformed $\mathbf{Q}_n$ are converted to (h, r, θ) cylindrical coordinates with the mapping:

$$h = q_z, \qquad r = \sqrt{q_x^2 + q_y^2}, \qquad \theta = \arccos(q_x/r).$$

The final step is to normalize h, r to be unit length on average.

4.1.3. Shape Distribution Computation

We divide the (h, r, θ) space into $d = m \times n \times p$ bins and calculate the histogram as follows:

$$H(k) = \frac{\sharp\{\mathbf{Q}_n : \mathbf{Q}_n \in bin(k)\}}{\sharp\{\mathbf{Q}_n\}},$$

where k denotes the kth bin of d bins and $\sharp\{\mathbf{Q}_n\}$ is equal to the number of points N. Thus, our shape descriptor becomes a d-vector $(H_1, H_2 ... H_d)$.

4.2. 3D Posture Recognition

The 3D human model we use in this application is composed of a sphere for the head, 2 cylinders (for the torso and for the neck) and 4 cones for the arms and legs. The model has 20 degrees of freedom (3 rotations for neck, 2 rotations for torso, upper arms and upper legs, 1 rotation for lower arms and lower legs, plus 3 global translations). Fig. 2 shows our human model in two simple postures. The representation of the model's shape context is achieved by first uniformly sampling the geometry surfaces and then computing the shape descriptor using the method presented above. To assist the human posture recognition procedure, a shape descriptor database is pre-generated by calculating the shape context for different joint angle configurations of the human model (e.g. rotating the upper arm around the Y axis in 4 steps and around the X axis in 9 steps). Human kinematics constraints are also applied in this step. Altogether, we generate about 2000 instances for different postures of the body.

4.2.1. Shape Similarity Measurement

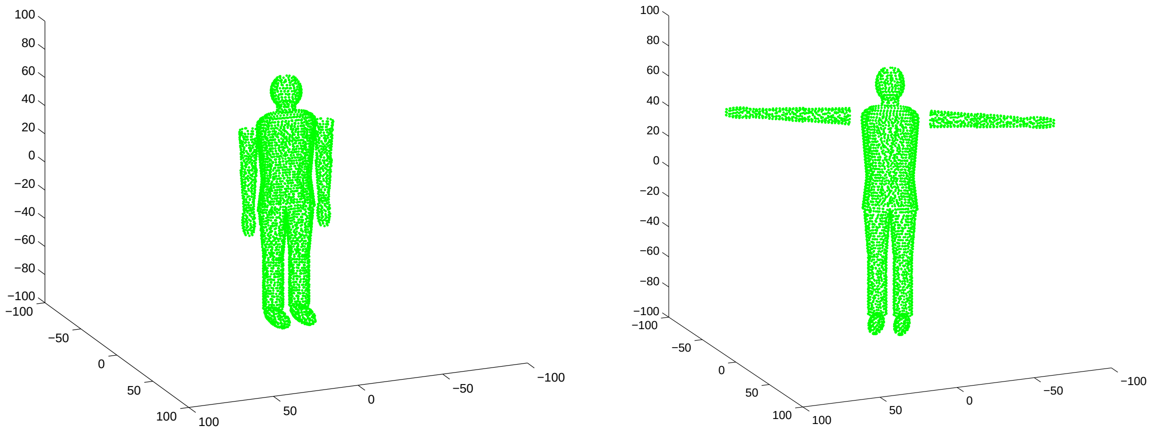
Given the shape descriptor of both observed human points cluster and human body model, the measurement of similarity could be defined as their Euclidean distance:

$$dist(H_m, H_o) = \sqrt{\sum_{i=1}^{d} (H_m(i) - H_o(i))^2},$$

where H_m and H_o denote shape descriptors of our human model and the observed points cluster, respectively. Since our modified cylinder frame based shape context descriptor is rotation invariant around the principal axis of the cylinder reference frame, the displacement $\Delta\theta$ between the model and real human points could induce a large dissimilarity. Here we propose a procedure to compensate for the ambiguity of the rotation displacement. The basic idea of this method is to search in θ space in each H_m (by rotating the shape descriptor of H_m along the principal axis) to achieve the minimum distance between $H_{m,\theta}$ and H_o. Hence, the equation of similarity measurement becomes:

$$dist(H_m, H_o) = \min_{\theta} \sqrt{\sum_{i=1}^{d} (H_{m,\theta}(i) - H_o(i))^2}, \tag{6}$$

where $H_{m,\theta}$ denotes the shape descriptor of the model rotated around the principal axis by θ. Our human gesture recognition procedure searches for the best match within the pre-generated shape descriptor database presented in the previous section. A nearest neighbor approach is applied to find the best match according to the modified distance function presented above.



(a) Standing (b) Arms spread

Figure 2. Models of two simple human postures.

5. RESULTS

Reconstruction results from a stereo image pair are shown in Fig. 3. Stereo reconstruction accuracy is evaluated by projecting 3D points back onto the right view. For all projected 2D points, their distance from all the edge points in the right image is computed. The edge pixel closest to a 2D point is obtained. The histogram of these distances is shown in Fig. 3(d). 80% of reconstructed points are off by less than 2 unit lengths. Although accuracy is fairly good, correspondence matching generally fails for edge pixels along the boundary of the object, meaning depth discontinuity will cause a shift of the background in different views, leading to a mismatch between window patches extracted from different views around these pixels.

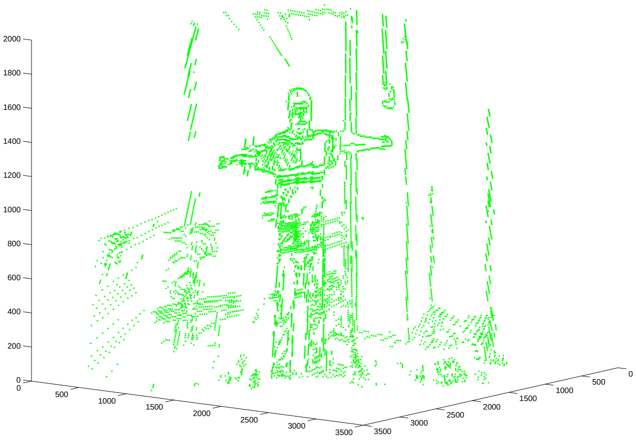
For shape matching, we use $d = 10\times10\times12$ bins in (h, r, θ) space. The histogram of measured 3D points is then compared with our database of shape descriptors generated from the model using Eqn. (6), which contains approximately 2000 postures. The $\Delta\theta$ used in the rotational alignment procedure is 30 degrees. Fig. 4(a) shows the similarity measure for 256 different upper body postures in the database. The best match corresponds to the minimum matching cost. Fig. 4(b) illustrates the 3D points cluster fit to the best matching model. Noise points can be observed around the torso, and some points of the hands and feet are missing due to reconstruction problems with these body parts. Since we use a global histogram to represent shape, however, these local variations have relatively little effect on overall posture recognition. The next-best matches also correspond to model configurations very similar to the best match (they only differ by the right upper arm's rotation).

6. CONCLUSIONS

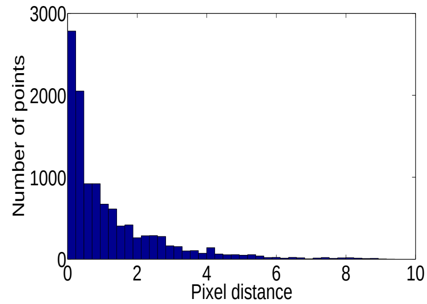
We have introduced the basic structure and components of our 3D surveillance system. In the current implementation, we have two pairs of stereo cameras and are detecting humans in the scene. Our reconstruction is accurate enough to separate the foreground from the background. We are able to reconstruct a standing human in the scene and recognize the person's posture. We are currently extending our system to more stereo camera pairs, which will enable us to improve reconstruction accuracy and will also better resolve occlusions in the scene. We are improving the matching models to make them more realistic. Our ultimate goal is to develop a surveillance system that can reconstruct and recognize human postures in real time.

(a) Left camera image

(b) Right camera image

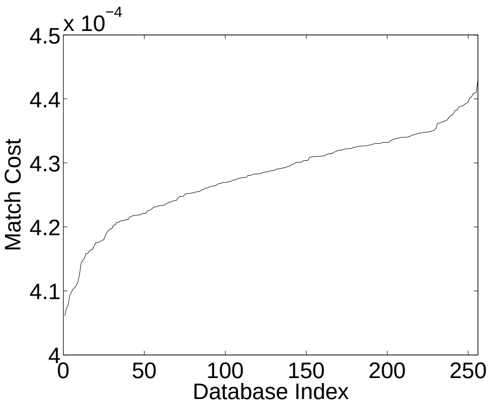


(c) 3D reconstructed scene

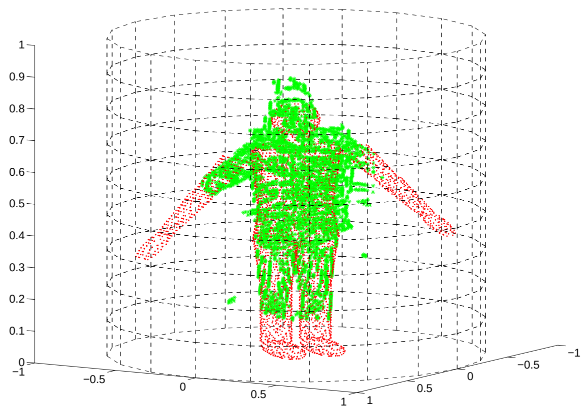


(d) Error histogram

Figure 3. 3D scene (c) generated from stereo pair (a,b) and histogram of reconstruction errors (d).



(a) Similarity measure for different postures



(b) Best match with reconstructed point cloud

Figure 4. 3D model matching with 3D point clusters.

REFERENCES

1. I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(8), pp. 809–830, 2000.
2. S. Huwer and H. Niemann, "Adaptive change detection for real-time surveillance applications," in *Proc. IEEE International Workshop on Visual Surveillance*, pp. 37–45, (Dublin, Ireland), 2000.
3. M. Agrawal, K. Konolige, and L. Iocchi, "Real-time detection of independent motion using stereo," in *Proc. IEEE Workshop on Motion and Video Computing*, pp. 207–214, (Breckenridge, CO, USA), 2005.
4. M. Harville, "Stereo person tracking with adaptive plan-view statistical templates," tech. rep., Hewlett-Packard, 2004.
5. D. Beymer and K. Konolige, "Real-time tracking of multiple people using continuous detection," in *Proc. International Conference on Computer Vision*, (Corfu, Greece), 1999.
6. K. Müller, A. Smolic, M. Drose, P. Voigt, and T. Wiegand, "3-D reconstruction of a dynamic environment with a fully calibrated background for traffic scenes," *IEEE Transactions on Circuits and Systems for Video Technology* **15**(4), pp. 538–549, 2005.
7. Y. I. Abdel-Aziz and H. M. Karara, "Direct linear transformation into object space coordinates in close-range photogrammetry," in *Proc. Symposium on Close-Range Photogrammetry*, pp. 1–18, (Urbana, IL, USA), 1971.
8. R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2004.
9. M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(8), pp. 993–1008, 2003.
10. L. Falkenhagen, "Hierarchical block-based disparity estimation considering neighbourhood constraints," in *Proc. International Workshop on SNHC and 3D Imaging*, pp. 115–122, (Rhodes, Greece), 1997.
11. S. Belongie and J. Malik, "Matching with shape context," in *Proc. IEEE Workshop on Content-based Access of Image and Video Libraries*, pp. 20–26, (Hilton Head, SC, USA), June 2000.
12. S. Belongie, J. Malik, and J. Puzhicha, "Efficient shape matching using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(11), pp. 1832–1837, 2005.
13. G. Mori and J. Malik, "Recovering 3D human body configurations using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(7), pp. 1052–1062, 2006.