Multi-feature graph-based object tracking

Murtaza Taj, Emilio Maggio, and Andrea Cavallaro

Queen Mary, University of London
Mile End Road, London E1 4NS (United Kingdom)
{murtaza.taj,emilio.maggio,andrea.cavallaro}@elec.qmul.ac.uk,
WWW home page: http://www.elec.qmul.ac.uk/staffinfo/andrea/

Abstract. We present an object detection and tracking algorithm that addresses the problem of multiple simultaneous targets tracking in real-world surveillance scenarios. The algorithm is based on color change detection and multi-feature graph matching. The change detector uses statistical information from each color channel to discriminate between foreground and background. Changes of global illumination, dark scenes, and cast shadows are dealt with a pre-processing and post-processing stage. Graph theory is used to find the best object paths across multiple frames using a set of weighted object features, namely color, position, direction and size. The effectiveness of the proposed algorithm and the improvements in accuracy and precision introduced by the use of multiple features are evaluated on the VACE dataset.

1 Introduction

Object tracking algorithms aim at establishing the correspondence between object observations at subsequent time instants by analyzing selected object features. This problem is usually divided into two major steps: the detection of foreground regions and the association of these regions over time.

A typical problem in the detection step is the definition of pre-processing and post-processing strategies under challenging lighting conditions, such as cast shadows, local and global illumination variations, and dark scenes. To address the problem of cast shadows, object and scene geometry, texture, brightness or color information can be used. Shadows can be modeled using Gaussians [1], multi-variate Gaussians [2] and mixture of Gaussians [3]. Texture analysis has also been used to detect shadows, based on the assumption that shadows do not alter the texture of the underlying surface [4]. A combination of features such as luminance, chrominance, gradient density and edges can also be used [5]. Moreover, edge and region information can be integrated across multiple frames [6]. Shadow color properties have also been used [7, 8, 3], based on the observation that a shadow cast on a surface equally attenuates the values of all color components. Although these methods succeed in segmenting shadows in a number of test sequences, they tend to fail when shadows are very dark. In this case, contextual information such as prior knowledge of object orientation can be used. In traffic monitoring, detecting the lanes of a road can improve the performance of shadow detection algorithms [9].

Once objects are detected, the second step aims at linking different instances of the same object over time (i.e., data association). A typical problem for data association is to disambiguate objects with similar appearance and motion. For this reason data association for object tracking can be assimilated to the motion correspondence problem. Several statistical and graph-based algorithms for tracking dense feature points have been proposed in the literature. Two methods based on statistics are the Joint Probabilistic Data-Association Filter [10] and the Multiple Hypotheses Tracking [11]. The major drawbacks of these two methods are the large number of parameters that need to be tuned and the assumptions that are needed to model the state space [12]. An example of graph-based method is Greedy Optimal Assignment [13], which requires a batch processing to deal with occlusions and detection errors, and assumes that the number of objects is constant over the time. A variable number of objects is allowed when dummy nodes are introduced in the graph in order to obtain a constant number of nodes per frame [14]. More elegant solutions have also been proposed: the best motion tracks are evaluated across multiple frames, based on a simple motion model. Next, node linking is performed after pruning unlikely motions [12]. Data association can also be performed by matching the blob contour using the Kullback-Leibler distance [15]. However, this method needs large targets to compute accurately the blob contour, and the correspondence is limited to two consecutive frames. Multi-frame graph matching [12] can be applied to motion correspondence using the appearance of regions around the points; then the global appearance of the entire object is computed with PCA over the point distribution [16]. Graph matching has also been used to find the object correspondence across multiple cameras [17] analyzing both color appearance and scene entry-exit object positions. Finally two-frame bipartite graph matching can be used to track objects in aerial videos based on gray level templates and centroid positions [18].

This paper proposes a tracking algorithm that copes with a variety of real-world surveillance scenarios, with sudden changes of the environmental conditions, and to disambiguate objects with similar appearance. To achieve these goals, the algorithm combines a statistical color change detector with a graph-based tracker that solves the correspondence problem by measuring the coherency of multiple object features, namely, color histograms, direction, position, and size. The video is equalized in case of dark scenes and the output of the background subtraction is post-processed to cope with shadows, global illumination changes caused by the passage of clouds and by vehicle headlights.

The paper is organized as follows. Section 2 describes the object detection algorithm. In Section 3 we present the graph matching strategy for multiple object tracking. Section 4 discusses the experimental results using different sets of features and validates the proposed approach on the VACE dataset [19]. Finally, Section 5 concludes the paper.

(a) (b) (c) (d)

Fig. 1. Contrast enhancement for improving object detection. (a) Reference frame; (b) current frame; (c) image difference before contrast enhancement; (d) image difference after contrast enhancement.

2 Object Detection

Foreground segmentation is performed by a statistical color change detector [20], a model-based algorithm that assumes additive white Gaussian noise on each frame. The noise amplitude is estimated for each color channel. Challenging illumination conditions typical of long surveillance videos, such as dark scenes, global and local illumination changes, and cast shadows need to be addressed separately.

Dark scenes are identified by analyzing the frame intensity distribution. A scene is classified as dark when more than 75% of the pixels in a frame are in the first quartile of the intensity range. In this case contrast and brightness are improved through image equalization.

Rapid global illumination changes are often associated to the passage of clouds. This results in large false positive detections, especially in regions in the shade of buildings or trees. To increase the contrast, the variance σ_0 of the difference image calculated between reference and first image should be similar to the variance σ_i of the difference between reference $I_{ref}(x, y)$ and current frame $I_i(x, y)$. Let β and ζ_0 be the brightness and the initial contrast, respectively; and let $\sigma_i = \sigma(\mid I_{ref}(x, y) - I_i(x, y) \mid)$. The contrast of the current difference image is modified at each iteration k using $\zeta_k = \zeta_{k-1} \pm s$ until the condition $|\sigma_{i,k} - \sigma_0| < \epsilon$ is satisfied. The pixel values Γ_k^j in the difference image are modified, for an 8-bit image, according to

$$\Gamma_k^j = \begin{cases} 0 & \text{if } a_k \cdot j + b_k < 0 \\ 255 & \text{if } a_k \cdot j + b_k > 255 \\ a_k \cdot j + b_k & \text{otherwise} \end{cases}, \tag{1}$$

where $j \in [1, 255]$ is the pixel value, $a_k = \frac{1}{1 - w \cdot \Delta_k}$, $b_k = a_k \cdot (\beta - \Delta_k)$, $w = 2/255$ and $\Delta = \frac{\zeta_k}{w \cdot \zeta_0}$. Fig. 1(d) shows a sample frame with increased contrast.

Vehicle headlights generate important local illumination changes. To address this problem, we perform an edge-based post-processing using selective morphology that filters out misclassified foreground regions by dilating strong foreground edges and eroding weak foreground edges. Next, 8-neighbor connected components analysis is performed to generate the foreground mask.

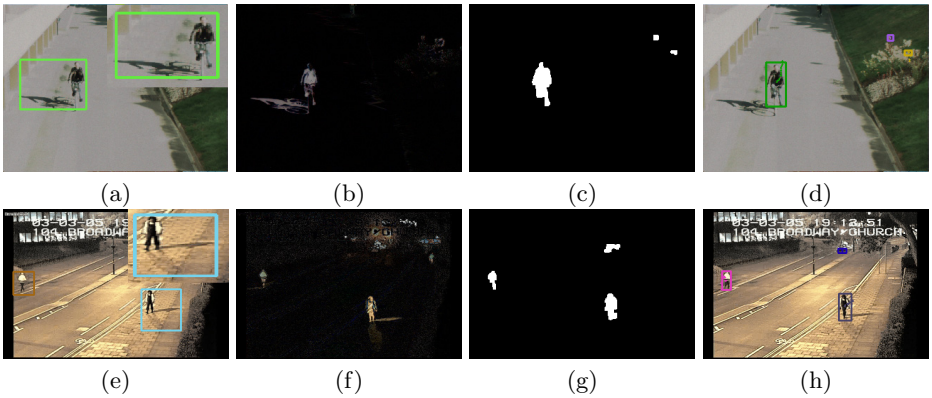

Fig. 2. Example of shadow removal to improve the accuracy of object detection. (a) Example of strong shadow; (b) difference image; (c) foreground mask after shadow segmentation; (d) final bounding box. (e) Example of multiple shadows; (f) difference image; (g) foreground mask after shadow segmentation; (h) final bounding box.

Finally, *cast shadows* are frequent local illumination changes in real-world sequences (Fig. 2(a), (e)) that affect the estimation of an object shape. Many surveillance scenarios are characterized by shadows that are too dark for a successful use of color-based techniques. For this reason, we use a model-based shadow removal approach that assumes that shadows are cast on the ground. Fig. 2 (c),(g) shows sample results of shadow removal.

The result of the object detection step is a bounding box for each blob (Fig. 2 (d),(h)). The next step is to associate subsequent detections of the same object over time, as explained in the next section.

3 Graph matching using weighted features

Data association is a challenging problem due to track management issues such as appearance and disappearance of objects, occlusions, false detections due to clutter and noisy measurements. Furthermore, data association has to be verified throughout several frames to validate the correctness of the tracks.

Let $\{X_i\}_{i=1\ldots K}$ be K sets of target detections, and $v(\mathbf{x}_i^a) \in V_i$ the set of vertices representing the detected targets at time i. Each $v(\mathbf{x}_i^a)$ belongs to D, a bi-partitioned digraph (i.e., a directional graph), such as the one reported in Fig. 3 (a). The candidate correspondences at different observation times are described by the gain g associated to the edges $e(v(\mathbf{x}_i^a), v(\mathbf{x}_j^b)) \in E$ that link the vertices. To obtain a bi-partitioned graph, a split of the graph $G = (V, E)$ is performed and two sets, V^+ and V^-, are created as copies of V. After splitting, each vertex becomes either a source (V^+) or a sink (V^-). Each detection $\mathbf{x}_i^a \in X_i$ is therefore represented by twin nodes $v^+(\mathbf{x}_i^a) \in V^+$ and $v^-(\mathbf{x}_i^a) \in V^-$ (Fig. 3 (c)). The graph is formed by iteratively creating new edges from the

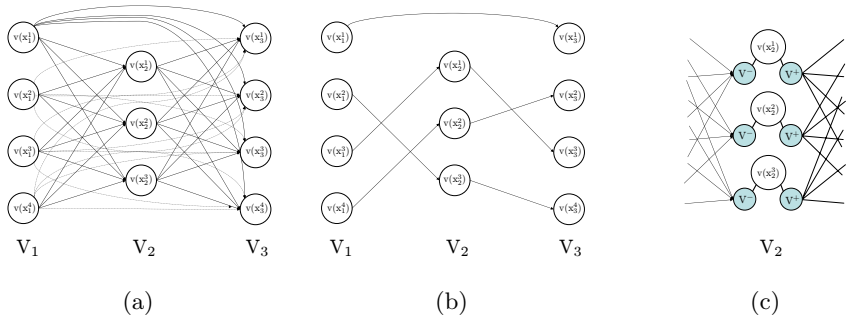

Fig. 3. Example of digraph D for 3 frames motion correspondence. (a) The full graph. (b) A possible maximum path cover. (c) Bi-partition of some nodes of the graph.

vertices $v^+(\mathbf{x}_i^a) \in V^+$ to the sink nodes $v^-(\mathbf{x}_K^b)$ associated to the new object observations X_K of the last frame.

Edges represent all possible track hypotheses, including miss detections and occlusions (i.e., edges between two vertices $v(\mathbf{x}_i^a)$ and $v(\mathbf{x}_j^b)$, with $j-i>1$). The best set of tracks is computed by finding the maximum weight path cover of G, as in Fig. 3 (b). This step can be performed using the algorithm by Hopcroft and Karp [21] with complexity $O(n^{2.5})$, where n is the number of vertices in G. After the maximization procedure, a vertex without backward correspondence models a new target, and a vertex without forward correspondence models a disappeared target. The depth of the graph K determines the maximum number of consecutive miss detected or occluded frames during which an object track can still be recovered. Note that despite larger values of K allow dealing with longer term occlusions, the larger the value of K, the higher is the probability of wrongly associating different targets.

The gain g between two vertices is computed using the information in X_i, where the elements of the set X_i are the vectors $\mathbf{x}_i^a$ defining $\mathbf{x}$, the state of the object:

$$\mathbf{x} = [x, y, \dot{x}, \dot{y}, h, w, H], \tag{2}$$

where (x, y) is the center of mass of the object, $(\dot{x}, \dot{y})$ are the vertical and horizontal velocity components, (h, w) are the height and width of the bounding box, and H is the color histogram. The velocity is computed based on the backward correspondences of the nodes. If a node has no backward correspondence (i.e., object appearance), then $\dot{x}$ and $\dot{y}$ are set to 0. The gain for each couple of nodes $\mathbf{x}_i^a, \mathbf{x}_j^b$ is computed based on the position, direction, appearance and size of a candidate target. The *position gain* g_1 based on the predicted and observed position of the point, is computed as

$$g_1(\mathbf{x}_i^a, \mathbf{x}_j^b) = 1 - \sqrt{\frac{[x_j^b - (x_i^a + \dot{x_i^a}(j-i))]^2 + [y_j^b - (y_i^a + \dot{y_i^a}(j-i))]^2}{{D_x}^2 + {D_y}^2}}, \tag{3}$$

where D_x and D_y are height and width of the image, respectively. Since the gain function is dependent on the backward correspondences (i.e. the speed at the previous step) the greedy suboptimal version of the graph matching algorithm is used [12]. The *direction gain* g_2 aims at penalizing large deviations in the direction of motion, is

$$g_2(\mathbf{x}_i^a, \mathbf{x}_j^b) = \frac{1}{2}\left(1 + \frac{(x_j^b - x_i^a)\dot{x_i^a}(j-i) + (y_j^b - y_i^a)\dot{y_i^a}(j-i)}{\sqrt{(x_j^{b\,2} + y_j^{b\,2})(x_i^{a\,2} + y_i^{a\,2})}}\right). \tag{4}$$

The *appearance gain* g_3 is the distance between color histograms of objects using the correlation method:

$$g_3(\mathbf{x}_i^a, \mathbf{x}_j^b) = \frac{\sum_{k=0}^{N}(H_{i,a}^{'}(k) \cdot H_{j,b}^{'}(k))}{\sqrt{\sum_{k=0}^{N}(H_{i,a}^{'}(k)^2 \cdot H_{j,b}^{'}(k)^2)}}, \tag{5}$$

where $H^{'}(k) = H(k) - \frac{1}{N \cdot \sum_{z=0}^{N} H(z)}$, N is number of histogram bins.

Finally, the *size gain* g_4 is the gain computed as absolute difference between the width and height of the objects represented by the nodes:

$$g_4(\mathbf{x}_i^a, \mathbf{x}_j^b) = 1 - \frac{1}{2}\left(\frac{\left|w_j^b - w_i^a\right|}{\max(w_j^b, w_i^a)} + \frac{\left|h_j^b - h_i^a\right|}{\max(h_j^b, h_i^a)}\right). \tag{6}$$

The *overall gain* g is a weighted linear combination of the position, direction, size and appearance gain as

$$g(\mathbf{x}_i^a, \mathbf{x}_j^b) = \alpha \cdot g_1(\mathbf{x}_i^a, \mathbf{x}_j^b) + \beta \cdot g_2(\mathbf{x}_i^a, \mathbf{x}_j^b) + \gamma \cdot g_3(\mathbf{x}_i^a, \mathbf{x}_j^b) + \delta \cdot g_4(\mathbf{x}_i^a, \mathbf{x}_j^b) - (j-i-1) \cdot \tau \tag{7}$$

where $\alpha + \beta + \gamma + \delta = 1$ and τ is a constant that penalizes the choice of shorter tracks. Since graph matching links nodes based on the highest weights, two trajectory points far from each other can be connected. To overcome this problem, gating is used and an edge is created only if $g > 0$.

4 Experimental results

We present experimental results on the VACE [19] dataset. The sequences are in CIF format at 25Hz. To evaluate the benefits introduced by different features, four configurations are compared: C-T, the baseline system with center of mass only; CB-T the system with center of mass and bounding box; CBD-T, the system with center of mass, bounding box and direction; and CBDH-T, the proposed system with all the previous features and the appearance model based on color histograms.

The parameters used in the simulations are the same for all scenarios. The change detector has $\sigma = 1.8$ and a kernel with $3x3$ pixels. A 32-bin histogram is used for each color channel. The weights used for graph matching are: $\alpha = 0.40$

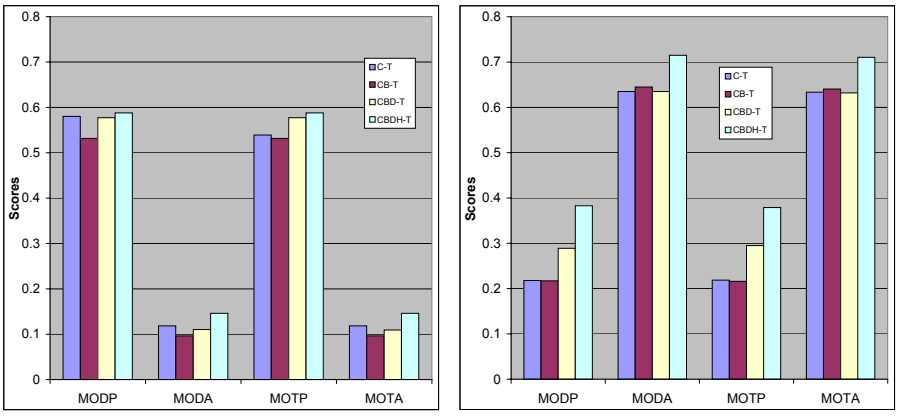

Fig. 4. Comparison of objective results using different set of features for detection and tracking on the Broadway/Church scenario, from the VACE dry run dataset (C-T: center of mass only; CB-T: center of mass and bounding box; CBD-T: center of mass, bounding box and direction; CBDH-T, the proposed system with all the previous features and the appearance model based on color histograms). Left: score for person detection and tracking. Right: score for vehicle detection and tracking.

(position), $\beta = 0.30$ (direction), $\gamma = 0.15$ (histogram), $\delta = 0.15$ (size), and $\tau = 0.043$. The objective evaluation is based on the 4 scores of the VACE protocol, namely Multiple Object Detection Accuracy (MODA), Multiple Object Detection Precision (MPDP), Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP) [19]. In order to use the VACE evaluation tool and the available ground truth, a simple pedestrian/vehicle classifier is added to the system, whose decision is based on the ratio of the width and the height of the bounding box, followed by a temporal voting mechanism.

Scores obtained with the different combinations of features are shown in Fig. 4. The results on the 4 scores show that the proposed algorithm (CBDH-T) produces a consistent improvement, especially in the case of vehicle tracking. This performance is not surprising as vehicles tend to have more distinctive colors than pedestrians. The use of direction as a feature improves detection and tracking precision more than detection and tracking accuracy (see Fig. 4 CBD-T vs. CB-T).

Sample tracking results for CBDH-T are shown in Fig. 5. Detected objects are identified by a color-coded bounding box, their respective trajectories and an object ID (top left of the bounding box). The results of the classification into one of the two classes, namely pedestrian (P) and vehicles (V), are shown on the top of the bounding box.

To conclude, in Fig. 6 we analyze the limits of the proposed algorithm. CBDH-T tends to merge tracks of small targets, such as vehicles far from the camera, when limited color information is available and the frame-by-frame motion direction is not reliable (Fig. 6 (a)). Another failure modality is due to the

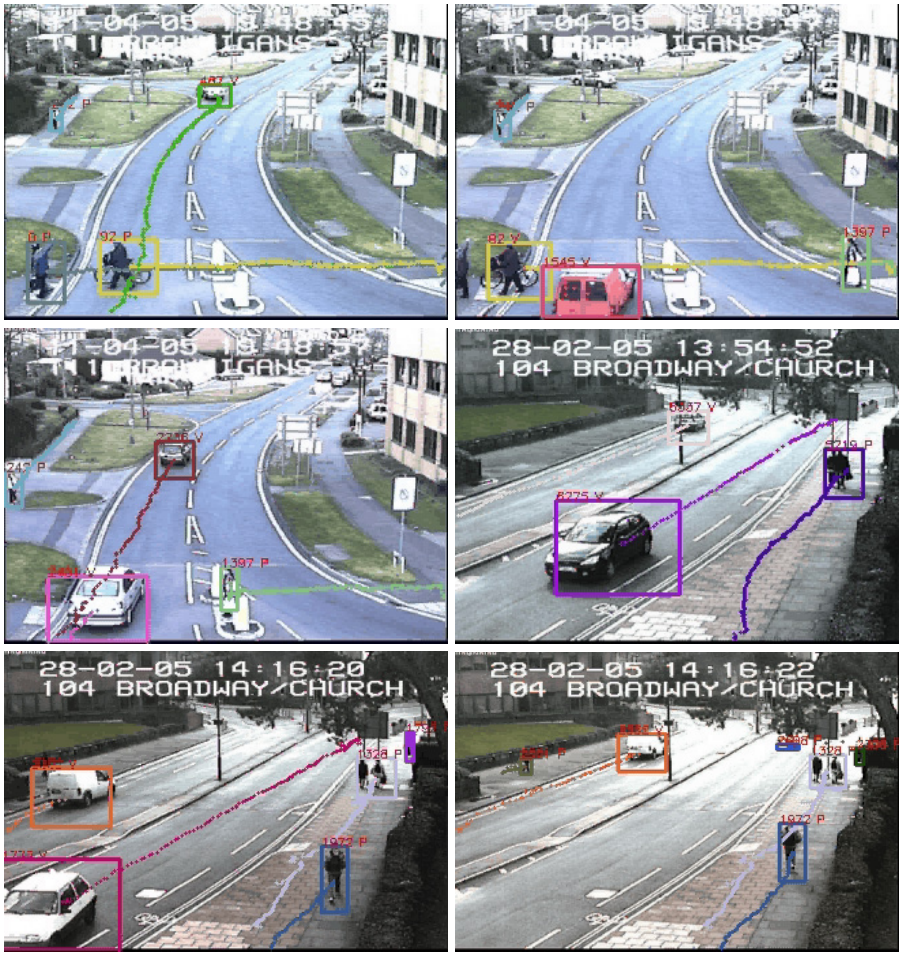

Fig. 5. Sample tracking results using the proposed detection and tracking algorithm (CBDH-T) on the VACE dataset.

foreground detector: when objects are too close to each other, such as pedestrians in groups or parked vehicles (Fig. 6 (b)), only one blob (i.e., one bounding box) is generated. We also noticed some instability in the detection of vehicles in dark scenes, due to variations in the illumination changes generated by the headlights. In Fig. 6 (d) the features used by the graph matching algorithm change drastically compared to Fig. 6 (c) because of a change in the object bounding box, thus generating an identity switch. A possible solution to both problems is to add to the system a detection algorithm based on prior knowledge (models) of the objects.

(a) (b) (c) (d)

Fig. 6. Examples of failure modes of the proposed algorithm. (a) Track ambiguity between two vehicles driving on opposite lanes far from the camera (see zoom). (b) Vehicles merged by the detector due to their proximity. (c),(d) Lost track due to variations in the object features caused by a significant change of the bounding box size (the two frames show the same vehicle at different time instants).

5 Conclusions

We presented a multiple object detection and tracking algorithm based on statistical color change detection and graph matching. The graph matching procedure uses multiple object features: position, color, size and direction. Experimental results showed that increasing the number of features and appropriately weighting them is an effective solution for improving tracking results in challenging real-world surveillance sequences, such as those of the VACE dataset. The algorithm demonstrated the ability to cope with changes in global illumination and local illumination conditions, using the same set of parameters throughout the dataset.

Future work includes the use of multiple views to increase the robustness of the detection and tracking algorithm and the integration of a state-of-the-art object classifier to improve the detection results.

References

1. Chang, C., Hu, W., Hsieh, J., Chen, Y.: Shadow elimination for effective moving object detection with gaussian models. In: Proc. of IEEE Conf. on Pattern Recog. Volume 2. (2002) 540–543
2. Porikli, F., Thornton, J.: Shadow flow: A recursive method to learn moving cast shadows. In: Proc. of IEEE International Conference on Computer Vision. Volume 1. (2005) 891–898
3. Martel-Brisson, N., Zaccarin, A.: Moving cast shadow detection from a gaussian mixture shadow model. In: Proc. of IEEE Conf. on Comp. Vis. and Pattern Recog. Volume 2. (2005) 643–648
4. Javed, O., Shah, M.: Tracking and object classification for automated surveillance. In: Proc. of the European Conference on Computer Vision, Copenhagen (2002)
5. Fung, G., Yung, N., Pang, G., Lai, A.: Effective moving cast shadow detection for monocular color image sequences. In: Proc. of IEEE International Conf. on Image Analysis and Processing. (2001) 404–409
6. Xu, D., Liu, J., Liu, Z., Tang, X.: Indoor shadow detection for video segmentation. In: IEEE Fifth World Congress on Intelligent Control and Automation (WCICA). Volume 4. (2004)

7. Huang, J., Xie, W., Tang, L.: Detection of and compensation for shadows in colored urban aerial images. In: IEEE Fifth World Congress on Intelligent Control and Automation (WCICA). Volume 4. (2004) 3098–3100
8. Salvador, E., Cavallaro, A., Ebrahimi, T.: Shadow identification and classification using invariant color models. In: Proc. of IEEE International Conf. on Acoustics, Speech, and Signal Processing. Volume 3. (2001) 1545–1548
9. Hsieh, J., Yu, S., Chen, Y., Hu, W.: A shadow elimination method for vehicle analysis. In: Proc. of IEEE Conf. on Pattern Recog. Volume 4. (2004) 372–375
10. Fortman, T., Bar-Shalom, Y., Scheffe, M.: Sonar tracking of multiple targets using joint probabilistic data association. IEEE J. Oceanic Eng. **8**(3) (1983) 173–184
11. Reid, D.: An algorithm for tracking multiple targets. IEEE Trans. Automat. Contr. **AC-24** (1979) 843–854
12. Shafique, K., Shah, M.: A noniterative greedy algorithm for multiframe point correspondence. IEEE Trans. Pattern Anal. Machine Intell. **27** (2005) 51–65
13. Veenman, C., Reinders, M., Backer, E.: Resolving motion correspondence for densely moving points. IEEE Trans. Pattern Anal. Machine Intell. **23**(1) (2001) 54–72
14. Rowan, M., Maire, F.: An efficient multiple object vision tracking system using bipartite graph matching. In: FIRA. (2004)
15. Chen, H., Lin, H., Liu, T.: Multi-object tracking using dynamical graph matching. In: Proc. of IEEE Conf. on Comp. Vis. and Pattern Recog. Volume 2. (2001) II–210–II–217
16. Mathes, T., Piater, J.: Robust non-rigid object tracking using point distribution models. In: British Machine Vision Conference, Oxford (2005)
17. Javed, O., Rasheed, Z., Shafique, K., Shah, M.: Tracking across multiple cameras with disjoint views. In: The Ninth IEEE International Conference on Computer Vision, Nice, France (2003)
18. Cohen, I., Medioni, G.G.: Detecting and tracking moving objects for video surveillance. In: CVPR, IEEE Computer Society (1999) 2319–2325
19. Kasturi, R.: Performance evaluation protocol for face, person and vehicle detection & tracking in video analysis and content extraction (VACE-II). Computer Science & Engineering University of South Florida, Tampa. (2006)
20. Cavallaro, A., Ebrahimi, T.: Interaction between high-level and low-level image analysis for semantic video object extraction. EURASIP Journal on Applied Signal Processing **6** (2004) 786–797
21. Hopcroft, J., Karp, R.: An $n^{2.5}$ algorithm for maximum matchings in bipartite graphs. SIAM J. Computing **2**(4) (1973) 225–230