A Spatio-Spectral Algorithm for Robust and Scalable Object Tracking in Videos

Alireza Tavakkoli[1], Mircea Nicolescu[2] and George Bebis[2,3]
[1] Computer Science Department, University of Houston-Victoria, Victoria, TX, USA
[2] Computer Science and Engineering Department, University of Nevada, Reno, NV, USA
[3] Computer Science Department, King Saud University, Riyadh, Saudi Arabia
tavakkolia@uhv.edu & (mircea,bebis)@cse.unr.edu

Abstract. In this work we propose a mechanism which looks at processing the low-level visual information present in video frames and prepares mid-level tracking trajectories of objects of interest within the video. The main component of the proposed framework takes detected objects as inputs and generates their appearance models, maintains them and tracks these individuals within the video. The proposed object tracking algorithm is also capable of detecting the possibility of collision between the object trajectories and resolving it without losing their models.

1 Introduction

Tracking of humans and other objects of interest within video frames is a very important task in many applications such as; video surveillance [1], perceptual user interfaces [2], and driver assistance [3]. Any reliable and robust object tracking mechanism may include two components [4]. The first component is responsible for generating and maintaining a model for the objects while the second process searches for potential new locations for these objects in the new frames. The target model generation deals with the dynamics of the tracked objects, learning of the scene priors and the evaluation of multiple hypotheses. The search components of the visual tracking mechanism mostly deals with the target representation localization and changes in the target appearance.

Shalom in [5] presents the filtering and data association process through a state space approach. The tracking given by the state space approach can be performed by an iterative Bayesian filtering [4]. Tthe Kalman filter and the Extended Kalman Filter (EFK) fail when applied to scenes with more clutter or when the background contains instances of the tracked objects. Through Monte Carlo based integration methods the particle filters [6] and the bootstrap filters [7] were proposed. Also in discrete state cases the Hidden Markov Models are used for tracking purposes in [8]. These methods do not provide reliable tracking results for non-rigid objects and deformable contours. The process of probability density propagation through sequential importance sampling algorithm, employed in particle filters, is computationally expensive.

The bottom-up approach to object tracking generates and maintains the target models and searches in new frames for their potential locations [4]. This approach assumes that the amount of changes in the location and appearance of the target is small. Tuzel *et al.*in [9] proposed a new non-statistical method under the target representation and localization by employing Lie algebra to detect and track objects under significant pose changes. Łoza *et al.*in [10] presented a structural similarity approach to object tracking in the video sequences. Recently SIFT features have been used in [11] tracking objects. However, reliable extraction and maintenance of the SIFT features and occlusion issues negatively affect this approach. Due to high computational cost - especially in cases where several object should be tracked over a long period of time- these methods fail to perform efficiently. Another issue in object tracking in video sequences is the ability to resolve occlusion. In this paper we propose an algorithm to reliable track multiple objects of interest and resolve possible occlusions which may occur as the number of tracked objects increase. The proposed approach is scalable to accommodate for increased number of objects within the field of view.

2 The Proposed Tracking Framework

The proposed algorithm is composed of two main stages.The first stage is the appearance correspondence mechanism. Once detected, photometric appearance based models are generated for the objects of interest. These models are considered to be the first degree estimation of the probability distribution of pixel colors. These models are then employed in the spatio-spectral connected component tracking mechanism to solve a correspondence problem between the set of detected connected components and the set of target appearance models. In the second phase, an occlusion resolution stage is used to halt the update process in case two or multiple targets occlude each other.

Since the appearance of the objects of interest are generally not known a priori, the only visual cue that can be used for detecting and tracking them is image motion. Our approach uses detected object from an efficient and reliable technique, based on background modeling and segmentation [12].

Figure 1 shows an overview of our proposed visual tracking algorithm. The proposed system uses the foreground masks detected by an object detector to generate connected components for each of the objects of interest. These connected components are then used to generate blobs for the objects which contain their spatial and spectral information. This information includes the height and width of the object, its center of mass, and the first degree statistical estimation of its photometric appearance. The algorithm looks for the possibility of objects occluding each other. We call this event a collision. If no collision is occurring in the scene, the spectral data association process is performed to solve the blob correspondence problem and track individual objects.

If the algorithm detects the possibility of the collision, a multi-hypothesis data association is performed to find the occluding object. Since the visible photometric information for current frame does not represent the occluded object(s),

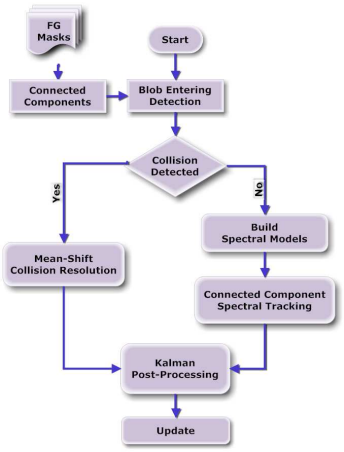

Fig. 1. The overview of the visual tracker using a spatio-spectral tracking mechanism.

their model will not be updated until the collision has been resolved. Since there will be only one occluding object, a simple kernel based tracking process will be used to track it. The blob information for this object is then retrained and updated for tracking purposes after the collision has been resolved.

In the final step of the algorithm, a simple Kalman filter is performed on the center of the blobs. Employing a Kalman filter in order to track individual points is an efficient process and does not complicate the computation requirements of the algorithm. This step helps refine the objects' tracking trajectories and remove the jitters in the trajectories that might occur during the object detection stage on the foreground region centers.

2.1 The visual tracking algorithm

We propose an efficient Spatio-Spectral Tracking module (SST) to track objects of interest in the video sequence. The detected foreground regions are processed further by employing a connected component processing in conjunction with a blob detection module to find objects of interest. These objects are tracked by their corresponding statistical models which are built from the objects' spectral (color) information. It is important to take not that the spatio-spectral coherency of tracked objects may be violated when two or more objects occlude each other.

A collision resolution mechanism is devised to address the issue of occlusion of objects of interest. This mechanism uses the spatial object properties such as their size, the relative location of their center of mass, and their relative orientations to predict the occlusion – i.e. collision.

Blob detection and object localization. In the blob detection module, the system uses a spatial connected component processing to label foreground regions. However, to label objects of interest a blob refinement framework is used to compensate for inaccuracies in physical appearance of the detected blobs

```
Maintain the list of tracking objects: O-L_{t-1}[1 : n]
For new frame t containing the foreground masks
   1. Detect the connected components
   2. Perform morphological smoothing to detect contingent objects
   3. Detect collision
      if no-collision:
   4. Maintain the new object list: CC-L_t[1 : k]
   5. if k > n determine if new objects are to be added to the new list
      if new objects then create = 1
      for i = 1 : k
      5.1. Generate the following:
               CC-L[i].Center
               CC-L[i].width
               CC-L[i].height
               CC-L[i].appearance
      for all unassigned O-L_{t-1} list objects
      5.2. find object O-L_{t-1}[j] : argmax [mean (p(CC-L_t[i]|O-L_{t-1}))]
      5.3. if probability is larger than threshold
               Assign: O-L_t[j] ← CC-L_t[i]
               Make object O-L_t[j] visible
           else
               Make object O-L_t[j] invisible
      5.4. if ( create = 1 )
               Assign: O-L_t[n + 1] ← CC-L_t[k]
               Make object O-L_t[n + 1] visible
   6. if collision:
      Maintain colliding object list: CO-L_t[1 : k]
      for colliding objects:
      6.1. find CO-L_t[j] : argmax [mean (p(CO-L_t[i]|O-L_{t-1}))]
      6.2. find maximum probability among colliding list objects
           suspend update for all the other objects in colliding list
           perform mean-shift tracking on the occluding object
   7. perform Kalman filter on the centers of visible objects
```

Fig. 2. The spatio-spectral object tracking algorithm.

due to unintended region split and merge, inaccurate foreground detection, and small foreground regions. A list of objects of interest corresponding to each detected blob is created and maintained to further process and track each object individually. This raw list of blobs corresponding to objects of interest is called the spatial connected component list. Spatial properties about each blob such as its center and size are kept in the spatial connected component list. The process of tracking individual objects based on their appearance and their corresponding spatial features is performed in the spatio-spectral tracking mechanism.

Spatio-spectral tracking (SST) mechanism. To track moving objects our proposed algorithm requires a model for individual objects. These "appearance models" are employed to search for correspondences among the pool of objects detected in new frames. Once the target for each individual has been found in the new frame they are assigned a unique ID. In the update stage the new information for only visible individual are updated.

Figure 2 show the pseudo-code of the proposed object tracking algorithm. Our modeling module represents an object with a set of statistical representation for its appearance. In the SST module a list of known objects of interest is maintained. During the tracking process the raw spatial connected component list is used as the list of observed objects. A statistical correspondence matching is employed to maintain the ordered objects list and track each object individually. The tracking module is composed of three components, appearance modeling, correspondence matching, and model update.

- ***Appearance modeling.*** Once each connected component is detected and processed their appearance models are generated. These appearance models

along with the objects location and first order geometric approximation produce an extended blob structure for the detected objects. In order to produce the geometric appearance of the detected objects, we use their corresponding connected components and geometric moments analysis. The 2-D geometric moments of a region are particularly important since they encode relevant visual and simple geometric features. In order to use these moments in computing geometric features of the objects in our work, we use the connected components. Along with these geometric features for the objects we extract orientation and their major and minor axis lengths. The objects'centers and their width are used in the process of collision detection.
The other component of the models of the objects in our algorithm is their photometric representation. Our current photometric appearance models are the first order statistical estimation of the probability density functions of pixel colors within the object.

– ***Correspondence matching.*** After the models are generated and objects are tracked in the previous frame at time $t-1$, a correspondence matching mechanism is employed in the current frame to keep track of the objects at time t. Unlike many target representation and localization methods our mechanism takes a better advantage of the object detection. The traditional approaches usually ignore the foreground objects and search in a neighborhood of the object in the previous frame to find the local maxima for the presence of the object.
Foreground objects generated using the connected component process from the foreground image populate a finite list of un-assigned objects in the current frame. We call this list $\{\text{CC-L}\}_t$ and the list of object appearance models from the previous frame $\{\text{O-L}\}_{t-1}$. The idea is to search on the unassigned list of objects their corresponding blob (appearance model) from the previous frame. Notice that in our algorithm instead of a spatial search over a window around each object and finding the best target match, we perform the search over the object list in the new frame. This decreases the computational cost of the algorithm compared to the traditional methods.
The proposed matching algorithm works by starting from the current frames connected component list. Let's denote the ith object from this list as; CC-$\text{L}_t(i)$. The algorithm goes through the object models from the $\{\text{O-L}\}_{t-1}$ list and finds the model which maximizes the likelihood of representing the CC-$\text{L}_t(i)$. If such model exists and is denoted by O-$\text{L}_{t-1}(j)$ then:

$$\text{O-L}_{t-1}(j) = \arg\max_k \left[mean\left(P\left(\text{C-L}_t(i)|\text{O-L}_{t-1}(k)\right)\right)\right] \qquad : \forall k \qquad (1)$$

Collision resolution. In order for the system to be robust to collisions – when objects get too close that one occludes the other – the models for the occluded individual may not reliable for tracking purposes. Our method uses the distance of detected objects as a means of detecting a collision. After a collision is detected we match each of the individual models with their corresponding representatives. The one with the smallest matching score is considered to be occluded. The occluded object's model will not be updated but its new position is predicted

```
For new frame t
   1. Calculate the speed of the objects
   2. for each object pair
      2.1. predict the new object centers in the
           next frame using Kalman filter
      2.2. if ( the two objects overlap ) then Collision = 1
      2.3. else Collision = 0
   3. return Collision
```

Fig. 3. The collision resolution algorithm.

by a Kalman filter. The position of the occluding agent is updated and tracked by a mean-shift algorithm. After the collision is over the spatio-spectral tracker resumes its normal process for these objects.

Figure 3 shows the algorithm which performs the collision detection and resolution. The collision detection algorithm assumes the center of the object and their velocity within consecutive frames are linearly related and the observation and measurement noises are normal with zero mean. Therefore, a Kalman filter can be used to predict the new locations of object centers. These information along with the width and height of the objects are used to predict the possibility of collision between multiple objects.

In our approach we assume that the discrete state of the objects is represented by their center of mass. Therefore, we have: $\mathbf{x}(k) = [C_x(k), \dot{C}_x(k), C_y(k), \dot{C}_y(k)]^T$, where $[C_x(k), C_y(k)]$ is the center of the object at frame k. Since the measurements are taken every frame at discrete time-steps with the rate of 30 frames per second, it is important to be able to predict whether the objects will collide given the observation and measurement parameters in the current frame. We assume that the object centers undergo a constant acceleration from frame to frame with unknown rates. We also assume that between each time-step the acceleration is randomly distributed as a normal probability density function with zero mean and an unknown covariance matrix. The governing equation that rules the relationship of consecutive states is given by: $\mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}a_k$, where a_k is the constant acceleration from time k to $k+1$, $\mathbf{G} = [1/2 \quad 1]^T$ is the acceleration vector and $\mathbf{F}$ is the the velocity matrix:

$$\mathbf{F} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Since we assume that the acceleration is drawn from random white zero mean noise, the state equation will become a linear function $\mathbf{h}$ affected by noise $\mathbf{n}$. Also we assume that the measurements are subject to a normal noise $\mathbf{v}$ which is independent from the observation noise. Therefore:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{h}(\mathbf{x}(k)) + \mathbf{n} \\ \mathbf{z}(k) &= \mathbf{f}(\mathbf{x}(k)) + \mathbf{v} \end{aligned} \tag{3}$$

where $n = \mathcal{N}(\mathbf{0}, \mathbf{Q})$ and $v = \mathcal{N}(\mathbf{0}, \mathbf{R})$. Since the state and measurement equations are linear and the noise is Gaussian, a Kalman filter can be used to predict

the location of the object centers in the new frames [13]. For each object pairs, a collision is about to occur if any of the following is true:

$$\begin{cases} C_1^{new} \cdot x < C_2^{new} \cdot x \Rightarrow C_1^{new} \cdot x + O_1 \cdot \frac{w_1}{2} \geq C_2^{new} \cdot x - O_2 \cdot \frac{w_2}{2} \\ C_1^{new} \cdot y < C_2^{new} \cdot y \Rightarrow C_1^{new} \cdot y + O_1 \cdot \frac{h_1}{2} \geq C_2^{new} \cdot y - O_2 \cdot \frac{h_2}{2} \\ C_1^{new} \cdot x > C_2^{new} \cdot x \Rightarrow C_1^{new} \cdot x - O_1 \cdot \frac{w_1}{2} \leq C_2^{new} \cdot x + O_2 \cdot \frac{w_2}{2} \\ C_1^{new} \cdot y > C_2^{new} \cdot y \Rightarrow C_1^{new} \cdot y - O_1 \cdot \frac{h_1}{2} \leq C_2^{new} \cdot y + O_2 \cdot \frac{h_2}{2} \end{cases} \quad (4)$$

where C_1 and C_2 are the center coordinates of each pair of objects and h and w is their respective height and width.

3 Experimental Results

In this section we compare the performance of the proposed technique in object tracking using several real video sequences that pose significant challenges. Also our algorithm's performance is compared with that of kernel-based tracking using Bhattacharyya Coefficient [4] with and without foreground detection and the structural-similarity measure of [10]. The performance evaluation consists of an evaluation of the frame rates of the systems the algorithms ability to resolve collisions and their robustness to changes in the objects' appearances.

Frame rate and tracking speed. As discussed in the previous sections the computational complexity of the proposed method is less than the existing object tracking algorithms. By limiting the search for potential targets to the linear list of detected objects in new frames, we decreased the search space. The advantage of this mechanism is the increased speed while tracking multiple objects. Figure 4 shows the performance of our method in terms of frame rate in comparison with two kernel based approaches as well as a particle filter algorithm.

Figure 4 (a) shows the results of our proposed tracking algorithm while Figure 4 (c)-(d) present the tracking results of the mean-shift algorithm [4], a mean-shift algorithm which uses the detected foreground object masks, and a particle filter with structural similarity measure [10], respectively. Our approach tracks the object with real-time performance of 20-25 frames per second (fps) while the single object tracking speed for the mean-shift algorithm is 7-10 fps (Figures 4 (b) and (c)). By introducing the second object to the scene the mean-shift speed drop to 4-7 fps compared to 20-25 fps in our algorithm. The particle filtering approach tracking time is more than one second (4 (d)).

Tracking performance in the presence of collision. Figure 5 presents the results of the proposed algorithm, the two mean-shift based methods, and the particle filter based algorithm on a video with two successive collisions. The figure shows two frames per collision, one taken before the collision and the other afterwards. Rows (a)-(d) represent the results of our approach, the mean-shift algorithm, the mean-shift algorithm with the foreground masks, and the particle filter technique, respectively. From the figure the proposed collision resolution mechanism in was able to effectively detect the collision and resolve it accordingly without the loss of object tracks. In this case the mean-shift and particle filter based methods -(b) and (d)- could also keep the tracking trajectories of the

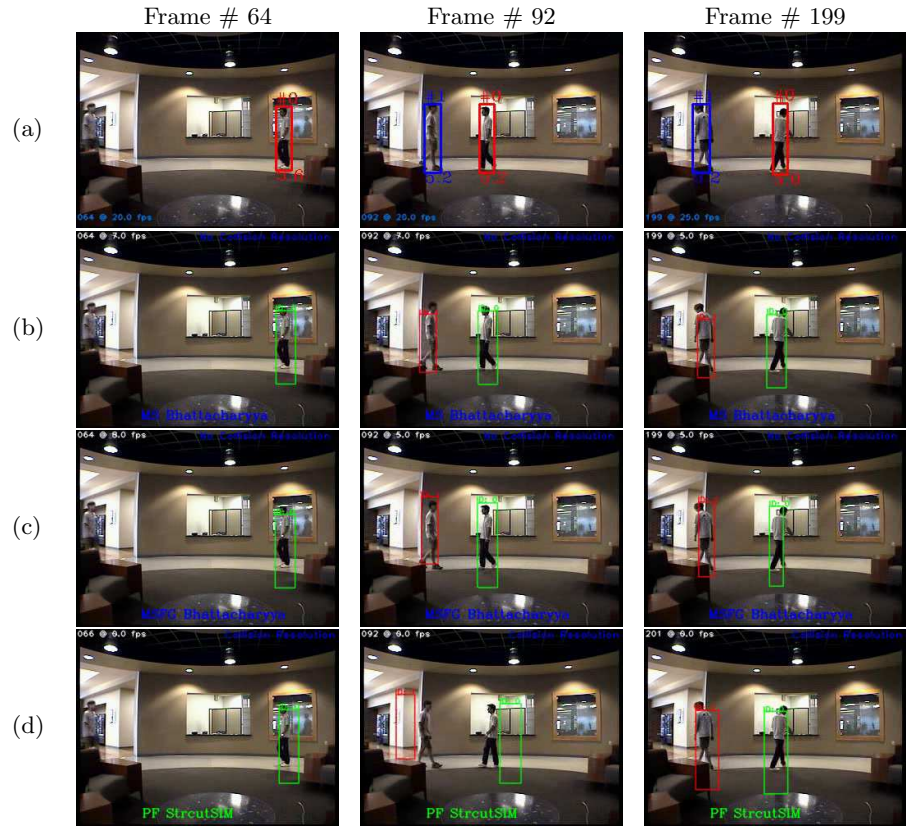

Fig. 4. Comparison of our visual object tracking algorithm with several traditional methods: (a) our method, (b) mean-shift algorithm, (c) mean-shift algorithm using foreground masks, (d) Particle Filter using structural similarity measure.

objects while the mean-shift algorithm which used the foreground regions -(c)- lost the track of occluding object.

By examining columns 3 and 4 we confirm that the proposed collision resolution mechanism in our approach was able to handle the second collision as well. However, the mean-shift algorithm in lost the track of the occluding object. Notice that the particle filter approach was robust to both collisions. However, as noted earlier this approach is very slow compared to our proposed technique and its tracking results are not as accurate as our algorithm.

Other challenging experiments. Figure 6 shows two challenging scenarios for visual object tracking algorithms which deal with target representation and localization. In Figure 6 three frames of videos taken in a dark lobby where lighting and reflective surfaces slightly change the objects' appearances in different locations. The proposed algorithm shows robustness to these challenge. The algorithm was also able to resolve the collision while the persons were passing.

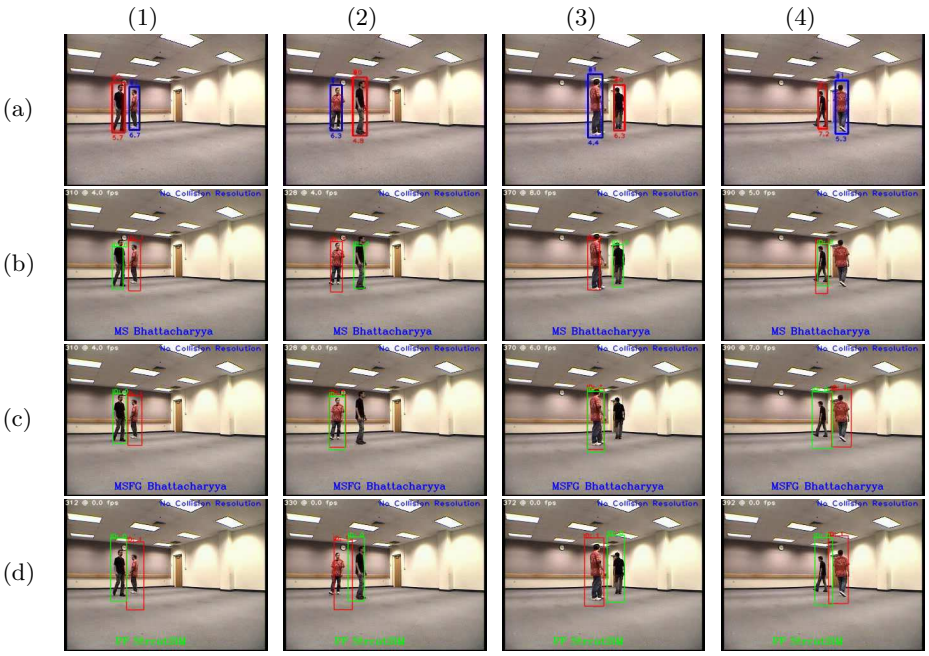

Fig. 5. Comparison of our tracking algorithm with several traditional methods in the presence of collision: (a) our method, (b) mean-shift algorithm, (c) mean-shift algorithm using foreground masks, and (d) Particle Filter using structural similarity measure.

4 Conclusions and Future Work

Based on the requirements of the real-time applications we proposed a non-parametric object tracking framework in this paper. Our approach takes advantage of the object detection process for tracking purposes. The detected objects are used to generate photometric and geometric appearance models. These appearance models are employed to reduce the target localization search space. The experimental evaluation indicate that our technique is faster than kernel-based approaches and shows more scalability. The performance of the proposed tracking algorithm is also compared to particle filter based methods. The results obtained from our technique showed superior performance over the traditional methods. In addition, A collision detection and resolution mechanism is introduced to our object tracking framework. This modules is responsible for predicting the possibility of collision between the foreground masks of two or more objects. The collision resolution mechanism is tested in several scenarios and the results show significant improvement over the kernel-based methods.

In our current implementation the photometric appearances of the objects are estimated by a single degree statistical model. Another future direction to this work is to introduce more complex and accurate models for the objects' appearances to achieve more accurate tracking results.

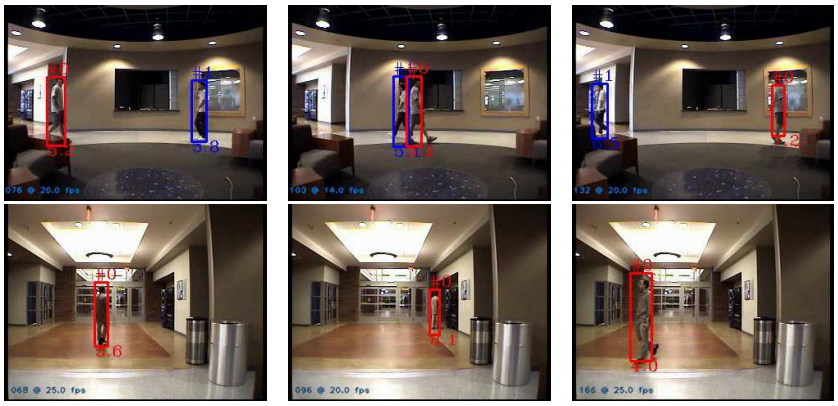

Fig. 6. The tracking results of the proposed visual object tracker under challenging conditions: (a) illumination changes, and (b) reflective surfaces.

References

1. Greiffenhagen, M., Comaniciu, D., Neimann, H., Ramesh, V.: Design, analysis and engineering of video monitoring systems: An approach and a case study. Proceedings of the IEEE **89** (2001) 1498–1517
2. Bradski, G.R.: Computer vision face tracking as a component of a perceptual user interface. IEEE Workshop on Applications of Computer Vision (1998) 214–219
3. Handman, U., Kalinke, T., Tzomakas, C., Werner, M., von Seelen, W.: Computer vision for driver assistance systems. Proceedings of SPIE **3364** (1998) 136–147
4. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Trans. Pattern Anal. Mach. Intell. **25** (2003) 564–575
5. Bar-Shalom, Y.: Tracking and data association. Academic Press Professional, Inc., San Diego, CA, USA (1987)
6. Kitagawa, G.: Non-gaussian state-space modeling of nonstationary time series. Journal of American Statistical Association **82** (1987) 1032–1063
7. Gordon, G., Salmond, D., Smith, A.: A novel approach to non-linear and non-gaussian bayesian state estimation. Proceedings of IEE **140** (1993) 107–113
8. Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of IEEE **77** (1989) 257–285
9. Tuzel, O., Porikli, F., Meer, P.: Learning on lie groups for invariant detection and tracking, Mineapolis, MN (2008) 1–8
10. Loza, A., Mihaylova, L., Bull, D., Canagarajah, N.: Structural similarity-based object tracking in multimodality surveillance videos. Mach. Vision Appl. **20** (2009) 71–83
11. Zhou, H., Yuan, Y., Westover, C.S.: Object tracking using SIFT features and mean shift. Computer Vision and Image Understanding **3** (2009) 345–352
12. Tavakkoli, A., Nicolescu, M., Bebis, G.: Efficient background modeling through incremental support vector data description. In Proceedings of the 19th International Conference on Pattern Recognition (2008)
13. Broida, T., Chellappa, R.: Estimation of object motion parameters from noisy images. Volume 8. (1986) 90–99