Multimed Tools Appl (2015) 74:227–243
DOI 10.1007/s11042-013-1846-5

Occlusion detection using horizontally segmented windows for vehicle tracking

Ahra Jo · Gil-Jin Jang · Bohyung Han

Published online: 11 January 2014

Abstract This paper proposes an efficient algorithm for detecting occlusions in a video sequences of ground vehicles using color information. The proposed method uses a rectangular window to track a target vehicle, and the window is horizontally divided into several sub-regions of equal width. Each region is determined to be occluded or not based on the color histogram similarity to the corresponding region of the target. The occlusion detection results are used in likelihood computation of the conventional tracking algorithm based on particle filtering. Experimental results in real scenes show that the proposed method finds the occluded region successfully and improves the performance of the conventional trackers.

Keywords Computer vision · Object tracking · Particle filters · Occlusion detection · Histogram similarity

1 Introduction

Tracking a moving object in a sequence of visual scenes is essential in a variety of applications such as image-based traffic surveillance systems, car safety alarms, and unmanned

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (No. 2012-0008090), and by the Converging Research Center Program through the Ministry of Science, ICT and Future Planning, Korea (No. 2013K000359).

A. Jo · G.-J. Jang (✉)
School of of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology (UNIST), Ulsan 689-798, Republic of Korea
e-mail: gijang.ucsd@gmail.com

A. Jo
e-mail: araya23@unist.ac.kr

B. Han
Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH), Pohang 790-784, Republic of Korea
e-mail: bhhan@postech.ac.kr

ground vehicles (UGV) [6, 8, 23, 37, 38, 40]. Video images contain measurement noise due to a variety of reasons, such as image sensor noise, vehicle movement, and other obstacles in the scene. Therefore, statistical approaches are usually adopted in solving the object tracking problems. One of the popular methods is a Kalman filter [5, 14, 20, 28, 30, 41]. It is assumed that the equations for the object location change are linear functions and their distributions are Gaussian, with additive, independent Gaussian noise. A state-space model is exploited to predict the object location in the observation from the past ones [28], and the prediction error is used to adaptively update the model. It has been applied to stereo camera-based object tracking [5]. However, a critical drawback of Kalman filters is that it fails to predict the observation when the change of movement is nonlinear and has non-Gaussian distribution. Particle filters based on Sequential Monte Carlo (SMC) methods were shown to be quite efficient in tracking visual objects [3, 25, 26, 29, 32, 33, 42], even when the dynamics of trajectories are non-linear and non-Gaussian. In particle filtering, the distribution of the position change vector is modeled by an ensemble of particles, and the object location is predicted by *maximum a posteriori* (MAP) estimation. The particles are usually updated by importance sampling where the importance is measured by the posterior probability given the observations [17, 18, 24]. Unlike Kalman filtering, the distribution is a non-parametric model stored in a set of particles, so any non-Gaussian dynamics can be properly approximated. Another type of popular methods is mean-shift algorithm [7, 9, 11, 27, 36]. A vincinity of the mean target location from the previous state is explored to predict the most likely object location. The observation is used to update the mean target location. Color histogram is usually employed to find the distance betwen the prediction and the observation. A joint, spatial color histogram is used in mean-shift tracking [10, 12, 38]. The advantage of mean-shift is that the search space is reduced, yielding a relatively fast tracking.

Those statistical approaches are very flexible and applicable to many realistic situations, but sometimes fail to track the target object especially when the target is occluded or cluttered. In the situations where the objects are completely occluded by other objects, they are handled by dynamic models such as a linear velocity model [5], and a nonlinear dynamic model [18]. In the cases of partial occlusion, the occluded regions or parts of contours are actively detection, and only reliable regions or contours are used in tracking. The shape priors of the target contours are given ahead of time using PCA (principal component analysis) [13], or constructed online [39]. Other methods include hierarchical decomposition [1], using *a priori* object shape information [16, 31, 34, 35], and learning the similarity patterns of occluded objects [21]. Although these methods have been shown effective, complicated parametric models and training data are usually required.

This paper proposes a novel method for identifying occluded part of the target object in an instantaneous scene for particle filtering-based visual object tracking. Particles, if modeled well, correspond to candidates for the next movement of the target object. In the proposed method they are defined by rectangular windows [25]. The proposed method is composed of two stages: occlusion detection in particles and occlusion pattern reasoning. In the first stage, the rectangular window obtained by each sample of the particle ensemble is divided horizontally into several non-overlapping, equi-sized sub-windows. The histogram distance of each sub-window to the corresponding part of the target window is computed, and used in determining if each sub-window is totally occluded. In the second stage, the sample occlusion detection results are combined to derive the most likely occlusion pattern. For each pixel of the current image in the sequence, the probability that the pixel belongs to the occluded region is computed by accumulating the sample occlusion detection results. The computed pixel occlusion probabilities are combined to identify which part of the target object is likely to be occluded by other objects. The occluded parts are excluded in computing the matching probability of each sample. The proposed method is well mingled in the particle filtering framework so that the tracking performance is not degraded even if there is no occlusion.

The paper is organized as follows. Section 2 describes the proposed method, Section 3 shows the experimental results in real car tracking examples, and Section 4 summarizes our findings and future research issues.

2 Method

2.1 Particle filter formulation

The particle filter is generally described by a standard state space model that has a set of unknown, hidden states linked to an observation process. There is the first-order Markovian assumption that the hidden state at time t is affected by the state at $t-1$ only. Given the observation sequence from the initial time 0 to t, denoted by $\mathbf{z}_{0:t} \doteq [\mathbf{z}_0 \ldots \mathbf{z}_t]$, target state $\mathbf{s}_t$ is a random vector whose behavior can only be described by *a posterior* probability given the observations, $p(\mathbf{s}_t|\mathbf{z}_{0:t})$, which is obtained by the following recursive probabilistic generation [25]:

$$\begin{aligned} & p(\mathbf{s}_t|\mathbf{z}_{0:t}) \propto \\ & \quad p(\mathbf{z}_t|\mathbf{s}_t) \int p(\mathbf{s}_t|\mathbf{s}_{t-1}) p(\mathbf{s}_{t-1}|\mathbf{z}_{0:t-1}) d\mathbf{s}_{t-1}\, , \end{aligned} \tag{1}$$

The conditional state density $p(\mathbf{s}_t|\mathbf{z}_{0:t})$ is then approximated by a set of M samples, $\left\{\mathbf{s}_t^m | m = 1 \ldots M\right\}$. The samples are called particles and the recursive derivation process in (1) within a sequential Monte Carlo framework is called particle filtering. To ignore samples with very low probabilities, an additional sampling based on some importance measure is generally employed [38].

2.2 Sample window segmentation for occlusion detection

In visual object tracking, a tracker is usually modeled by a rectangular region in the observed image, called a window. Assuming that the initial target position is known, the spatial information of the window is usually defined by a vector of center coordinate and scale from the initial window [25, 38]. Those constitute a sample in the particle filtering. The similarity of the target and each sample of particle filters can be measured by the resemblance of the color histograms. If other interfering objects blocks the target object partly or entirely, it may disrupt the histogram of the sample and lower the similarity significantly.

To identify which part of the target is occluded, we horizontally divide each sample window into several sub-windows with equal width. As shown in Fig. 1, the candidate tracking window is horizontally divided into N non-overlapping sub-windows,

$$\mathcal{W}(\mathbf{s}_t^m) = \bigcup_{i=1}^{N} \mathcal{W}_i(\mathbf{s}_t^m). \tag{2}$$

The color information of each sub-window is extracted by 110 histogram bins using the hue-saturation-value (HSV) color space [19, 25], then only 80 % bins out of 110 are chosen based on the probabilistic palette model [15]. Let $q_i(k; \mathbf{s}_t^m)$ and $q_i^*(k)$ be functions returning the relative frequency at histogram bin k measured from $\mathcal{W}_i(\mathbf{s}_t^m)$ and the same sub-window

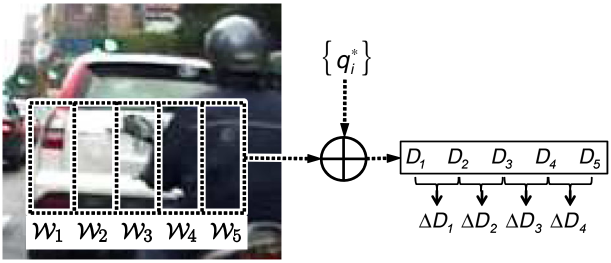

$\{q_i^*\}$
$\mathcal{W}_1$ $\mathcal{W}_2$ $\mathcal{W}_3$ $\mathcal{W}_4$ $\mathcal{W}_5$
D_1 D_2 D_3 D_4 D_5
ΔD_1 ΔD_2 ΔD_3 ΔD_4

Fig. 1 Division of a sample window into five horizontal sub-windows

in the initial target image, respectively, then the distance from the initial target is calculated by the Bhattacharyya distance [11]:

$$D_i(\mathbf{s}_t^m) = \left[1 - \sum_{k=1}^{K} \sqrt{q_i^*(k) q_i(k; \mathbf{s}_t^m)} \right]^{1/2}, \tag{3}$$

where K is the total number of histogram bins. Because motor vehicles move on the ground, an interfering object intrudes from the side and passes by the target, so horizontal division is likely to find the various patterns of partial occlusions.

Between every pair of adjacent sub-windows, the forward difference for the sub-window distance is computed:

$$\Delta D_i = D_{i+1} - D_i\ ,\ i = 1, \ldots, N-1, \tag{4}$$

where the argument $\mathbf{s}_t^m$ in (3) is omitted for a compact notation. The value of ΔD_i represents the magnitude and direction of the local change in the histogram distance from $\mathcal{W}_i$ to $\mathcal{W}_{i+1}$. A positive, large ΔD_i is observed when there is a big distance leap between adjacent sub-windows, which is the case that $\mathcal{W}_{i+1}$ is occluded and $\mathcal{W}_i$ is not. A negative ΔD_i leads to the opposite situation. The use of forward difference instead of absolute distance enables eliminating the effect of the overall distance elevation due to illumination change or other color-influencing factors.

Figure 2a–d illustrate the horizontal window segmentation for occlusion detection. The rectangles around the target vehicles are tracking windows divided by five sub-windows. The first bar graphs below the images represent the histogram distances of sub-windows, and the second bar graphs display their forward differences. In Fig. 2a and b, the target is the white vehicle, and a motorcycle blocks the target in different regions. In Fig. 2a the sub-window distances $D_3 \sim D_5$ are much larger than D_1 and D_2, resulting in the largest ΔD_2. In this case, the indices of the occluded sub-windows are $\{3, 4, 5\}$. In Fig. 2b, the occluded region is identified by $\{1, 2\}$ because the magnitude of ΔD_2 is large enough to the negative side. From these, the left occlusion boundary may be found by sub-window i if ΔD_i is negatively large enough, and the right boundary is found by $i+1$ if ΔD_i is positively large enough. Figure 2c has both left and right boundaries at $i = 2$ and 3, so the occluded sub-window indices are $\{2, 3\}$. Figure 2d also has both at sub-windows 4 and 1, but there are two interfering objects from the outside because the left index is larger than the right one.

In Table 1, we propose an algorithm to classify various occlusion patterns using the forward difference of sub-window histogram distances. The output of the algorithm is, for each sample, a set of integers in $[1,\ N]$ for the indices of the occluded sub-windows. Out of $\Delta D_1 \sim \Delta D_{N-1}$, only the maximum and the minimum are considered to prevent unreliable

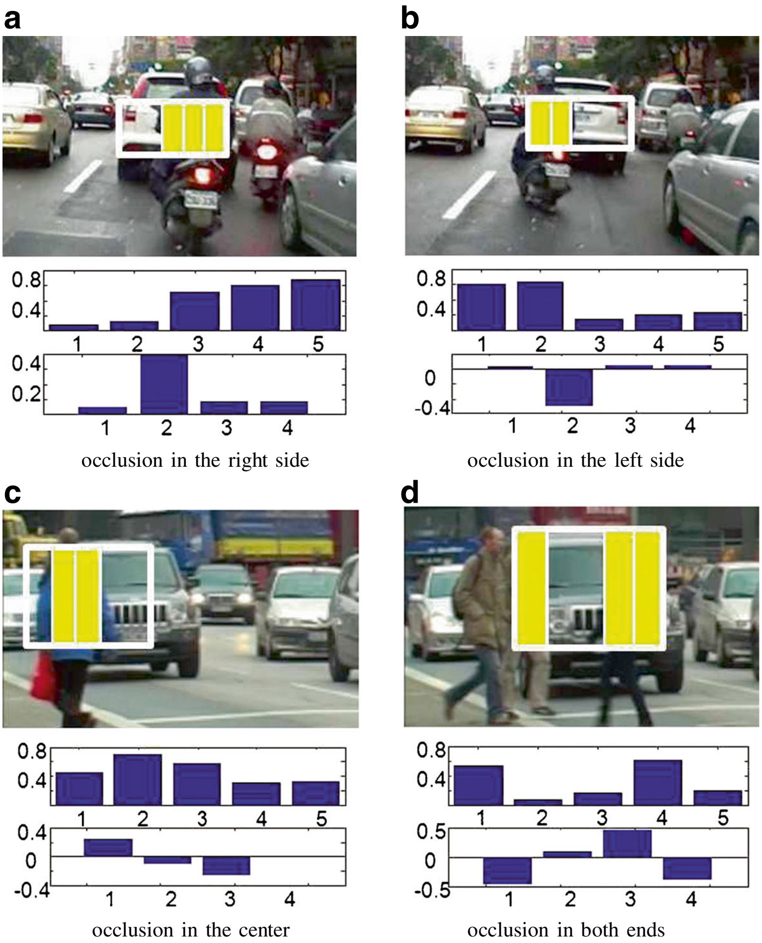


Fig. 2 Various occlusion patterns found by horizontal split. The indices of occluded sub-windows are: **a** {3, 4, 5}. **b** {1, 2}. **a** {2, 3}. **a** {1, 4, 5}

occlusion boundaries. In lines 7–10, the left and right occlusion boundaries are found by i^+ and i^-. With an appropriate choice of $\theta_{\Delta D}$, the occlusion patterns of individual samples are correctly identified. Lines 12 and 14 correspond to the cases of Fig. 2c and d, respectively. However, when $i^+ = 1$ and $i^- = N$, it can be either total or no occlusion. The two cases are distinguished by comparing the average histogram distance with a threshold θ_D as shown in line 17. The range of the Bhattacharyya distance is [0, 1], so 0.5 was a good start for $\theta_{\Delta D}$ and θ_D.

2.3 Finding a global occlusion pattern

Figure 3 illustrates is overall finding global occlusion pattern method. In A, the occluded sub-window indices are found by Table 1. Let $T(x, y, \mathbf{s}_t^m)$ be a function whose value

Table 1 Occlusion pattern reasoning for individual samples

```
$I_o(\mathbf{s})$: a set of occluded sub-windows indices for sample $\mathbf{s}$
$\theta_{\Delta D}$: cutoff value for determining occlusion boundary
$\theta_D$: cutoff value for determining total occlusion

For each sample $\mathbf{s}_t^m$,
   compute $\{D_i\}$ and $\{\Delta D_i\}$
   assign $i^+ \leftarrow \arg\max\{\Delta D_i\} + 1$
   assign $i^- \leftarrow \arg\min\{\Delta D_i\}$
   if $|\Delta D_{i^+-1}| < \theta_{\Delta D}$, $i^+ \leftarrow 1$
   if $|\Delta D_{i^-}| < \theta_{\Delta D}$, $i^- \leftarrow N$

   if $i^+ \leq i^-$,
     $I_o(\mathbf{s}_t^m) \leftarrow \{i \mid i^+ \leq i \leq i^-\}$
   else
     $I_o(\mathbf{s}_t^m) \leftarrow \{i \mid 1 \leq i \leq i^- \cup i^+ \leq i \leq N\}$

   if $i^+ = 1$, $i^- = N$, and $\frac{1}{N}\sum_{i=1}^{N} D_i < \theta_D$,
     $I_o(\mathbf{s}_t^m) \leftarrow \emptyset$
end of loop
```

is 1 when a pixel (x, y) belongs to a window defined by a sample $\mathbf{s}_t^m$ in (2), such that

$$T(x, y, \mathbf{s}_t^m) = \begin{cases} 1 & \text{if } (x, y) \in \mathcal{W}(\mathbf{s}_t^m), \\ 0 & \text{otherwise} \end{cases}. \tag{5}$$

The score function that a pixel (x, y) belongs to the target region at time t is obtained by averaging $T(x, y, \mathbf{s}_t^m)$ over all the samples,

$$T(x, y, t) = \frac{1}{M}\sum_{m=1}^{M} T(x, y, \mathbf{s}_t^m). \tag{6}$$

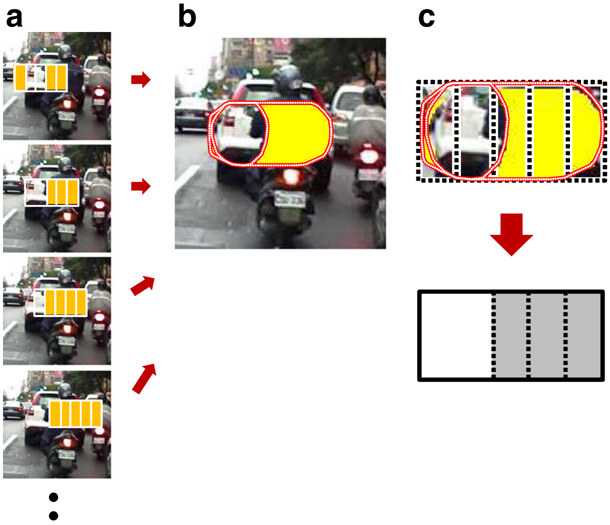


Fig. 3 Process of detecting occlusion region using the occlusion map

Similarly, let $O(x, y, \mathbf{s}_t^m)$ be an indicator function that a pixel (x, y) belongs to any occluded sub-window defined by sample $\mathbf{s}_t^m$:

$$O(x, y, \mathbf{s}_t^m) = \begin{cases} 1 & \text{if } (x, y) \in \mathcal{W}_i(\mathbf{s}_t^m)\,,\ \exists i \in I_o(\mathbf{s}_t^m) \\ 0 & \text{otherwise} \end{cases}, \tag{7}$$

where a set of indexes of the occluded sub-windows, $I_O(\mathbf{s}_t^m)$, is defined by the algorithm in Table 1.

The score function that a pixel (x, y) is occluded is obtained by averaging $O(x, y, \mathbf{s}_t^m)$ over all the samples, such that

$$O(x, y, t) = \frac{1}{M} \sum_{m=1}^{M} O(x, y, \mathbf{s}_t^m)\,. \tag{8}$$

Using the score function, an object target region and occluded parts can be found by simple thresholding. Let $\mathcal{R}_T(t)$ be a target object region and $\mathcal{R}_O(t)$ be an occlusion region obtained by

$$\mathcal{R}_T(t) = \{(x, y) | T(x, y, t) > \theta_T\}\,, \tag{9}$$

$$\mathcal{R}_O(t) = \{(x, y) | O(x, y, t) > \theta_O\}\,, \tag{10}$$

where θ_T and $\theta_O \in [0, 1]$ are fixed threshold values. Please note that $O(x, y, \mathbf{s}_t^m)$ is always less than or equal to $T(x, y, \mathbf{s}_t^m)$, so $O(x, y, t) \leq T(x, y, t)$ for any pixel (x, y). Therefore, we enforce $\theta_O > \theta_T$ to make $\mathcal{R}_O(t) \subset \mathcal{R}_T(t)$.

The procedure in (2)–(10) is illustrated in Fig. 3A and B. The region surrounded by the outside contour is the target region, $\mathcal{R}_T(t)$, and the lightly colored region inside the contour is the global occlusion region, $\mathcal{R}_O(t)$. Then, for a practical reason, the free-form occlusion pattern $\mathcal{R}_O(t)$ is converted to a sub-window-form occlusion patterns. In Fig. 3C, a window $\mathcal{W}^*(t)$ is found by a smallest rectangle to surround all $\mathcal{R}_T(t)$, and it is divided horizontally in the same way as (2), $\mathcal{W}^*(t) = \bigcup_i \mathcal{W}_i^*(t)$.

Each sub-window $\mathcal{W}_i^*(t)$ is determined to be occluded or not individually. The ratio of the number of pixels in the occlusion area to that in the target area, written by

$$\gamma(t, i) = \frac{A(\mathcal{R}_O(t) \cap \mathcal{W}_i^*(t))}{A(\mathcal{R}_T(t) \cap \mathcal{W}_i^*(t))}\,, \tag{11}$$

where $A(\cdot)$ is an area function of image regions. Then, let I_t^* be a index set of sub-windows of the target region that are likely to be unoccluded, which is given by

$$I_t^* = \{i | \gamma(t, i) \leq 0.5\}. \tag{12}$$

The global occlusion pattern is combined results of occluded sub-windows. This process expressed in the bottom of Fig. 3C.

The distance of sample $\mathbf{s}_t^m$ from the initial target is updated by I_t^*:

$$D(I_t^*, \mathbf{s}_t^m) = \frac{\sum_{i \in I_t^*} D_i(\mathbf{s}_t^m)}{N}. \tag{13}$$

Finally, the likelihood of a single sample in (1) is obtained by

$$p(\mathbf{z}_t | \mathbf{s}_t^m) \propto \exp\left[-\lambda \{D^2(I_t^*, \mathbf{s}_t^m)\}\right], \tag{14}$$

where λ is a positive contant that is empirically determined.

3 Experimental results

The performance of the proposed method is compared with the conventional particle filtering [26], L1 tracker [4] and mean-shift tracking algorithm [7, 11, 27, 36]. We test three video sequences capturing real ground vehicles with occlusions. The videos include various types of occlusions that usually happen with motor vehicles and pedestrians.

3.1 Evaluation method

To compare the performances quantitatively, we used the normalized intersection ratio [22, 29] and the tracking error measure [29]. The normalized intersection ratio is computed by the ratio of overlapped are between hand-labeled ground truth and the tracking window from the tracker state, given by

$$accuracy = \frac{|\mathcal{G} \cap \mathcal{T}|}{|\mathcal{G} \cup \mathcal{T}|}, \tag{15}$$

where $\mathcal{G}$ and $\mathcal{T}$ are sets of pixels within ground truth and tracker output regions, respectively, and the cardinality operator $|\cdot|$ returns the number of pixels in a set [22].

We added another performance measure for measuring the general tracking error, computed by the Euclidean distance between the center points of ground truth and the tracking window from the tracker state [29].

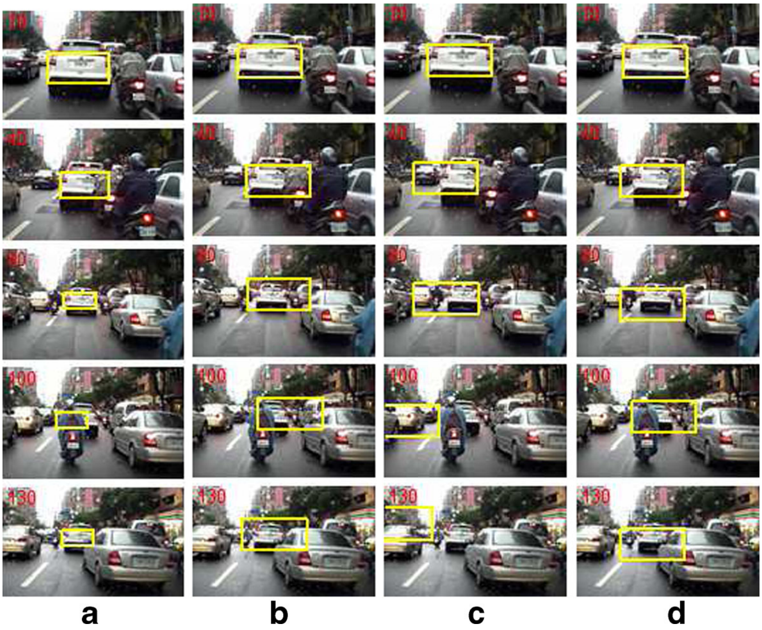


Fig. 4 Comparison of the tracking performances for motorcycle occlusion sequence. **a** ours, **b** L1 tracker, **c** conventional particle filtering, and **d** mean-shift

3.2 Occlusion between cars and motorcycles

Figure 4 shows the tracking results of the proposed method and the conventional methods. The scene is from a downtown area, and the target is the white vehicle in the center, which

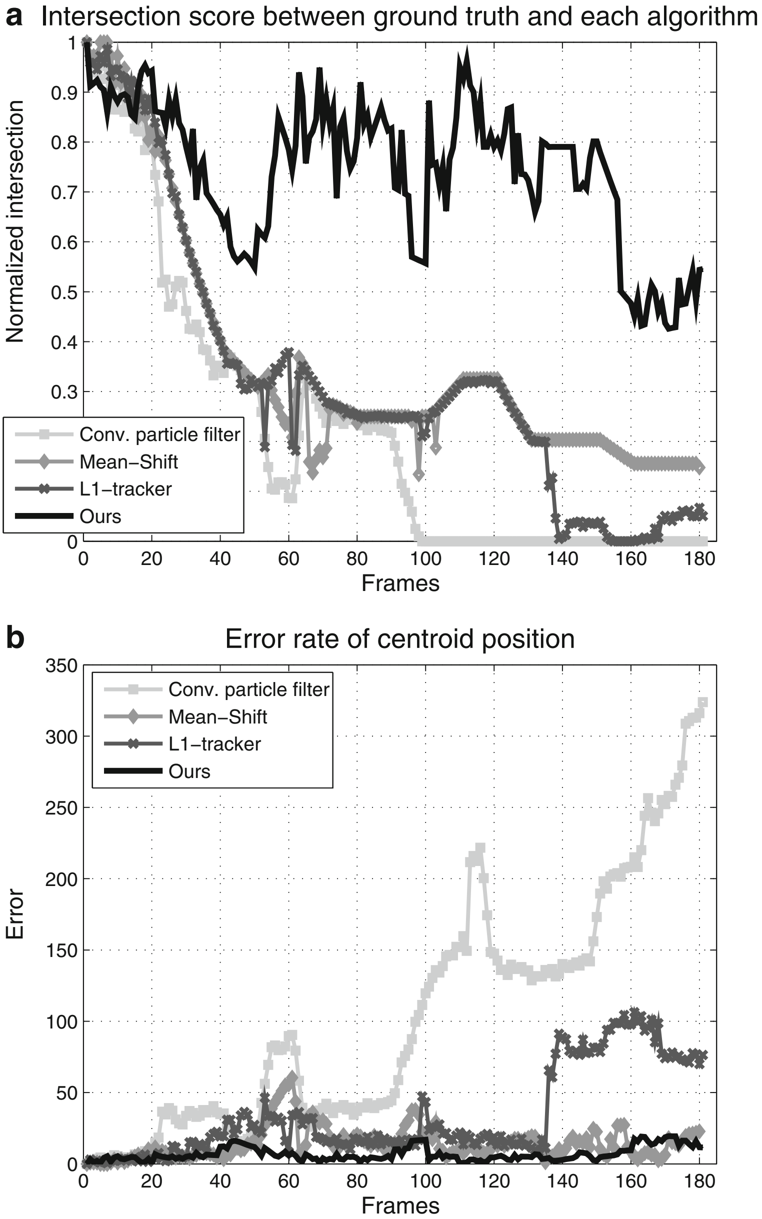


Fig. 5 Comparison of the tracking performances for motorcycle occlusion sequence. **a** intersection ratio, **b** tracking error by centroid distance between ground truth and the tracker window

is occluded by two motorcycles. The motorcycles move very fast and partly block the target vehicle in the video sequences while passing through the gap between the target and the observer. The first column, Fig. 4a, shows the tracker windows by yellow boxes for the proposed method. The other columns, Fig. 4b–d, show the results of L1 tracker [4], basic particle filtering [26], and mean-shift tracking [11], respectively. In the first row (frame 10), there no occlusion, and the four methods successfully tracks the target. In the second row (frame 40), a motorcycle driver blocks the target to the right. The proposed method and L1 tracker keep track of the target including the occluded area. However, the particle filters is shifted to the left due to the occlusion, and mean-shift tracker is enlarged because of the uncertainty added by the occlusion. In the third row (frame 80), although there is no occlusion, all three conventional methods exhibit incorrect tracking windows, enlarged to include the target. This is because of the tracking error accumulated from the past frames. At frame 100, another motorcycle hugely occluded the target, and this time even the proposed method failed to correctly track the target. Unlike the other 3 methods, the particle filters totally loses the target due to the occlusion. At frame 130, a little occlusion by another car from the right-bottom corner, and the proposed method restored to a correct position, L1 and mean shift trackers track the target with enlarged tracking windows. However, the particle filters cannot restore the error, so it perfectly lost the target. When occlusion occurs, the likelihood of the target is decreased, so the uncertainly grows. Generally L1 and mean-shift trackers extend the tracking window to recover the uncertainty. The proposed method actively finds the uncertain parts due to the occlusion and subtracts from the likelihood calculation, so even the occluded part of the target was able to be tracked.

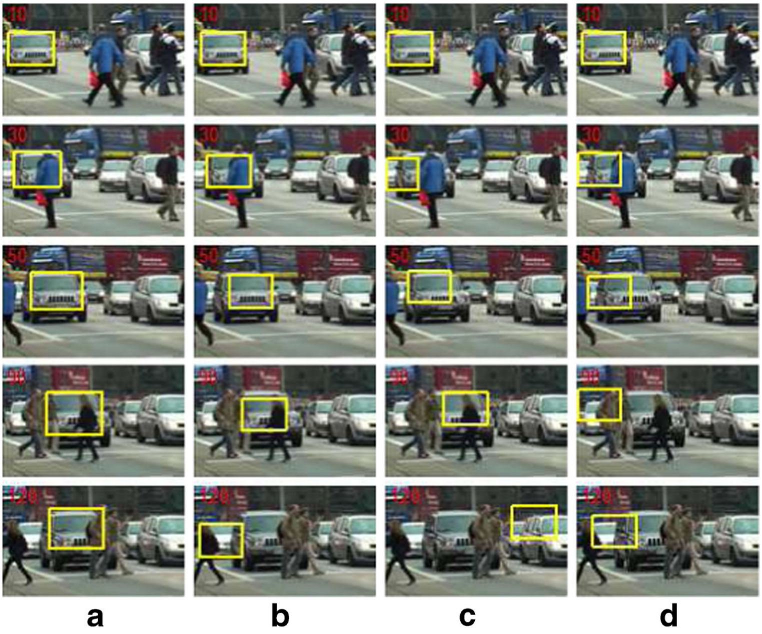


Fig. 6 Comparison of the tracking performances for the “TUD-Crossing” sequence. **a** ours, **b** L1 tracker, **c** conventional particle filtering, and **d** mean-shift

A quantitative evaluation was carried out using intersection ratio and tracking error by the center positions introduced in Section 3.1. The evaluation results are shown in Fig. 5. The top graph shows the intersection ratios, ranging from 0 (no intersection) to 1 (perfect match). Larger the number, better the tracking result is. A few frames before 40, occlusion

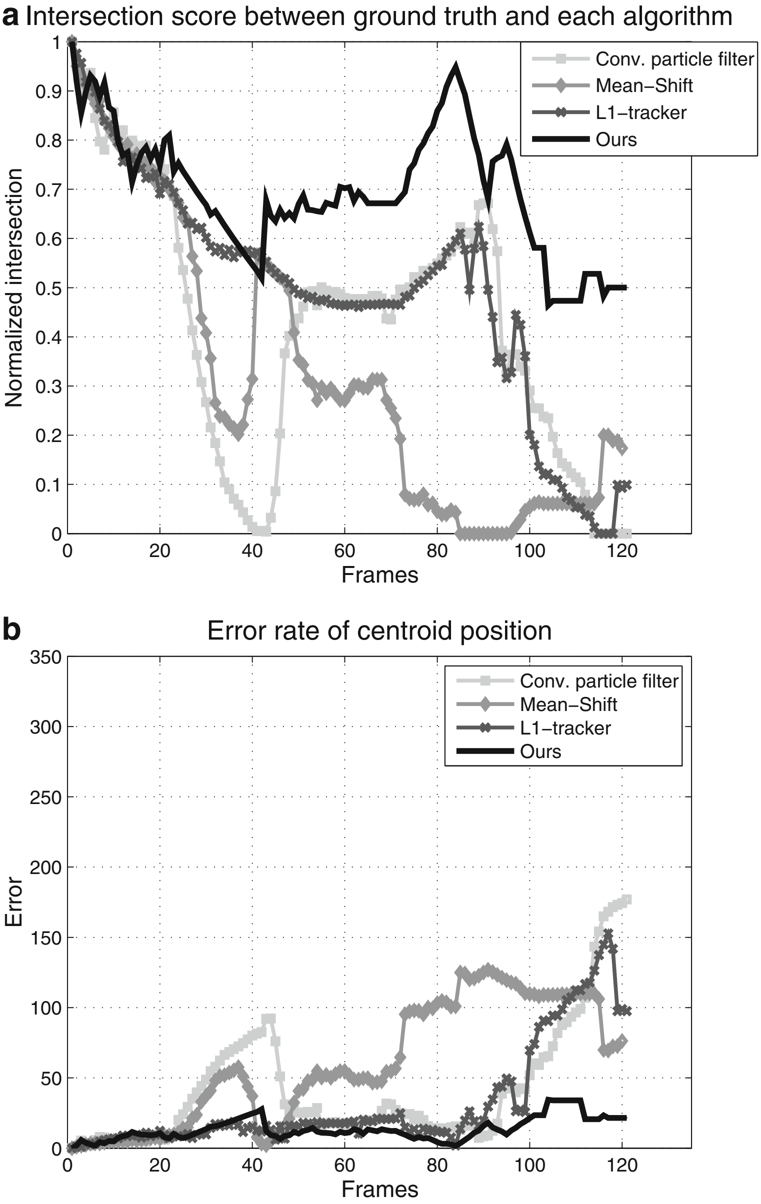


Fig. 7 Comparison of the tracking performances for the "TUD-Crossing" sequence. **a** intersection ratio, **b** tracking error by centroid distance between ground truth and the tracker window

occurs as shown in the second row of Fig. 4. The tracking errors are accumulated until frame 50, and the proposed method recovered the intersection ratio to larger than 0.8 around frame 60, while the others could not recover and have the intersection ratio below 0.4. Another huge occlusion occurs around frame 100, and the particle filters completely lost the target, and the intersection ratio became 0. L1 and mean-shift tracker kept enlarged tracking window and the ratio value stayed almost the same. The proposed method was affected by the occlusion at frames 95–100, the ratio value being below 0.6, but quickly recovers to 0.8. The same phenomena were observed in the centroid distance error measure. The quantitative comparison proves that the proposed method remarkably improved the tracking performance when occlusion occurs.

3.3 Occlusion by pedestrians

Two more vehicle tracking scenarios containing occlusions were also tried. The images in Fig. 6 are from "TUD-Crossing" dataset [2], where a lot of pedestrians pass before the target vehicle. Figure 7a and b are the quantitative comparison by the intersection ratios and centroid distances. In Fig. 6a, the proposed method successfully tracks the target vehicle even when pedestrians block the target. Especially at frames 90–120, 3 pedestrians pass before the target, and the tracking performance is not affected. In Fig. 6c, particle filtering without occlusion detection, the pedestrians hugely affected the tracker, and the target is lost perfectly at frame 120. Figure 6b and d shows the results of L1 and mean-shift trackers. Although they keep track of the target even with the occlusion, the performance is greatly decreased due to the pedestrian occlusion.

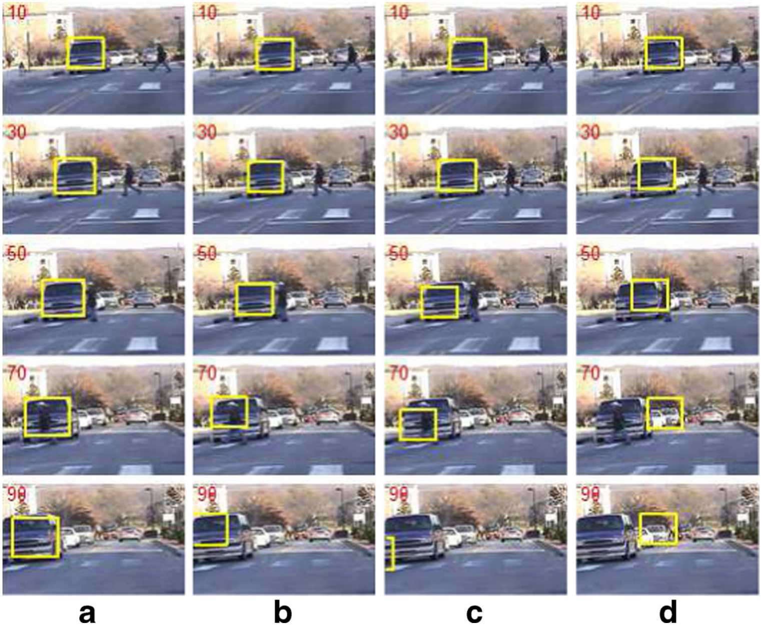


Fig. 8 Comparison of the tracking performances for a single pedestrian crossing sequence. **a** ours, **b** L1 tracker, **c** conventional particle filtering, and **d** mean-shift

Figure 8 is another pedestrian-vehicle case. The tracking windows are drawn by yellow rectangles, and the quantitative performance comparison is done in Fig. 9. Similarly to the previous results, the proposed method outperformed the conventional methods a lot.

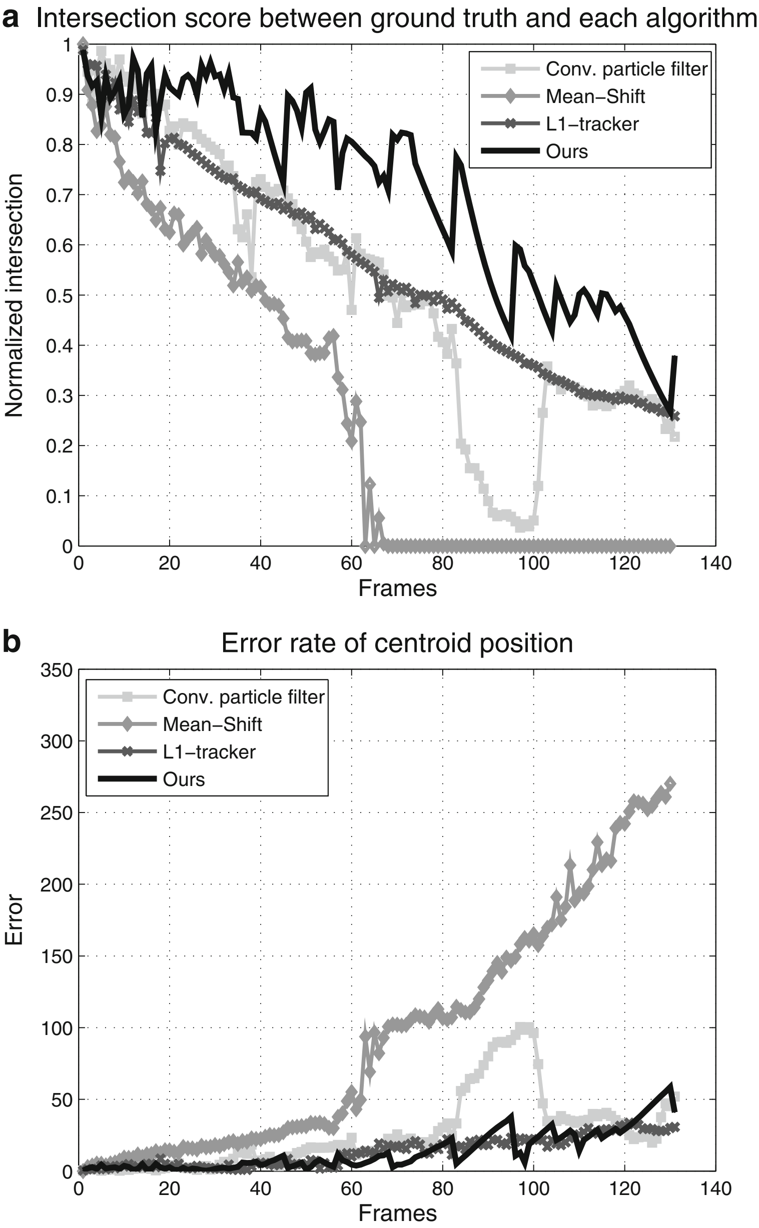


Fig. 9 Comparison of the tracking performances for a single pedestrian crossing sequence. **a** intersection ratio, **b** tracking error by centroid distance between ground truth and the tracker window

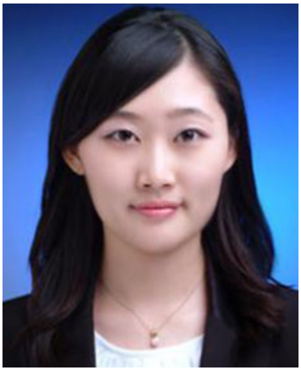
4 Conclusions

This paper proposes a practical algorithm for detecting occlusions in a color image sequence of ground vehicles based on matching color histograms of horizontally segmented rectangular windows. The proposed method divides the tracking windows into a number of vertical windows, and each sub-window is determined to be occluded or not to find a unified occlusion pattern. The unified occlusion pattern is then used to update the likelihood of the current tracking window and the target window, and the target can be tracked including the occluded region, while other methods cannot include the occluded region in the tracking windows. Performance comparison on three images sequences of vehicles and pedestrians shows the validity of the proposed method. Future work include applying other types of image descriptors based on the shape of the target to improve the performance of the proposed occlusion detection.

References

1. Ablavsky V, Sclaroff S (2011) Layered graphical models for tracking partially occluded objects. IEEE Trans Pattern Anal Mach Intell 33:1758–1775
2. Andriluka M, Roth S, Schiele B (2008) People-tracking-by-detection and people-detection-by-tracking. In: CVPR08, pp 1–8
3. Arulampalam S, Maskell S, Gordon N, Clapp T (2002) A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking. IEEE Trans Signal Process 50(2):174–189
4. Bao C, Wu Y, Ling H, Ji H (2012) Real time robust l1 tracker using accelerated proximal gradient approach. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 1830–1837
5. Beymer D, Konolige K (1999) Real-time tracking of multiple people using continuous detection. In Proceedings of the 7th international conference on computer vision, Kerkyra, Greece
6. Bouttefroy PLM, Bouzerdoum A, Phung S, Beghdadi A (2009) Vehicle tracking using projective particle filter. In: IEEE international conference on advanced video and signal based surveillance (AVSS), pp 7–12
7. Cheng Y (1995) Mean shift, mode seeking, and clustering. IEEE Trans Pattern Anal Machine Intell 17:790–799
8. Chu C-T, Hwang J-N, Pai H-I, Lan K-M (2011) Robust video object tracking based on multiple kernels with projected gradients. In: IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 1421–1424
9. Comaniciu D, Meer P (1999) Mean shift analysis and applications. In: Proceedings of the 7th international conference on computer vision, Kerkyra, Greece, pp 1197–1203
10. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Machine Intell 24(5):603–619
11. Comaniciu D, Ramesh V, Meer P (2000) Real-time tracking of non-rigid objects using mean shift. In: Proceedings of the IEEE conference on computer vision and pattern recognition, vol II. Hilton Head, pp 142–149
12. Comaniciu D, Ramesh V, Meer P (2003) Kernl-based object tracking. IEEE Trans Pattern Anal Machine Intell 25(5):564–575
13. Cremers D, Kohlberger T, Schnorr C (2002) Non-linear shape statistics in mumford-shah based segmentation. In: Proceedings of the European conference on computer vision, Copenhagen, Denmark
14. Dellaert F, Thorpe C (1998) Robust car tracking using Kalman filtering and Bayesian templates. In: Proceedings of SPIE, pp 72–83
15. Foley JD, van Dam A, Feiner SK, Hughes J (1990) Computer graphics: principles and practice, 2nd edn. Addison-Wesley
16. Hu W, Zhou X, Hu M, Maybank S (2009) Occlusion reasoning for tracking multiple people. IEEE Trans Circ Syst Video Technol 19:114–121

17. Isard M, Blake A (1998) Condensation—Conditional density propagation for visual tracking. Intl J Comput Vis 29(1):5–28
18. Isard M, MacCormick J (2001) Bramble: a bayesian multiple-blob tracker. In: Proceedings of the 8th international conference on computer vision, Vancouver, Canada, pp 34–41
19. Jojic N, Caspi Y (2004) Capturing image structure with probabilistic index maps. In: Proceedings of the IEEE conference on computer vision and pattern recognition, vol 1. Washington DC. pp 212–219
20. Kalman RE (1960) A new approach to linear filtering and prediction problems. Trans Am Soc Mech Eng D J Basic Eng 82:35–45
21. Kwak S, Nam W, Han B, Han JH (2011) Learning occlusion with likelihoods for visual tracking. In: Proceedings of the 14th international conference on computer vision, Barcelona, Spain, pp 1551–1558
22. Kwak S, Nam W, Han B, Han JH (2011) Learning occlusion with likelihoods for visual tracking. In: 2011 IEEE international conference on computer vision (ICCV), pp 1551–1558
23. Lee K-H, Hwang J-N, Yu J-Y, Lee K-Z (2013) Vehicle tracking iterative by kalman-based constrained multiple-kernel and 3-d model-based localization. In: IEEE international symposium on circuits and Systems (ISCAS), pp 2396–2399
24. MacKay DJC (1998) Introduction to Monte Carlo methods. In: Learning in graphical models. Kluwer Academic Press, pp 175–204
25. Perez P, Hue C, Vermaak J, Gangnet M (2002) Color-based probabilistic tracking. In: Proceedings European conference on computer vision, vol I. Copenhagen, Denmark, pp 661–675
26. Perez P, Vermaak J, Blake A (2004) Data fusion for visual tracking with particle filter. Proc IEEE 92(3):495–513
27. Quast K, Kaup A (2013) Shape adaptive mean shift object tracking using gaussian mixture models. In: Adami N, Cavallaro A, Leonardi R, Migliorati P (eds) Lecture notes in electrical engineering, analysis, retrieval and delivery of multimedia content, vol 158. Springer, New York, pp 107–122
28. Salarpour A, Salarpour A, Fathi M, Dezfoulian M (2011) Vehicle tracking using Kalman filter and features. Signal Image Proc Int J (SIPIJ) 2:1–8
29. Scharcanski J, de Oliveira AB, Cavalcanti PG, Yari Y (2011) A particle-filtering approach for vehicular tracking adaptive to occlusions. IEEE Trans Veh Technol 60:381–389
30. Stenger B, Mendonça PRS, Cipolla R (2001) Model-based hand tracking using an unscented kalman filter. In: Proceedings of the British machine vision conference, vol I. Manchester, UK, pp 63–72
31. Sudderth EB, Mandel MI, Freeman WT, Willsky AS (2005) Distributed occlusion reasoning for tracking with nonparametric belief propagation. In: Advances in neural information processing systems, pp 1369–1376
32. Tanizaki H (2000) Nonlinear and non-gaussian state-space modeling with Monte Carlo techniques: a survey and comparative study. North-Holland
33. Tanizaki H, Mariano RS (1998) Nonlinear and non-gaussian state-space modeling with Monte-Carlo simulations. J Econ 83(1–2):263–290
34. Wu B, Nevatia R (2007) Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. Intl J Comput Vis 75(2):247–266
35. Wu Y, Yu T, Hua G (2003) Tracking appearances with occlusions. In: CVPR03, vol 1. pp 789–795
36. Xiaojing Zhang CS, Yue Y (2013) Object tracking approach based on mean shift algorithm. J Multimedia 8(3):220–225
37. Xiong T, Debrunner C (2004) Stochastic car tracking with line- and color-based features. IEEE Trans Intell Transp Syst 5(4):324–328
38. Yilmaz A, Javed O, Shah M (2006) Object tracking: a survey. ACM Comput Surv 38:263–290
39. Yilmaz A, Li X, Shah M (2004) Contour based object tracking with occlusion handling in video acquired using mobile cameras. IEEE Trans Pattern Anal Machine Intell 26(11):1531–1536
40. Zhang Z, Huang K, Tan T, Wang Y (2010) 3d model based vehicle tracking using gradient based fitness evaluation under particle filter framework. In: international conference on pattern recognition (ICPR), pp 1771–1774
41. Zhong J, Sclaroff S (2003) Segmenting foreground objects from a dynamic textured background via a robust kalman filter. In Proceedings of the 9th international conference on computer vision. Nice, France, pp 44–50
42. Zhou S, Chellappa R, Moghaddam B (2004) Visual tracking and recognition using appearance adaptive models in particle filters. IEEE Trans Image Process 11:1434–1456



Ahra Jo is a graduate student at Ulsan National Institute of Science and Technology (UNIST), South Korea. She received her B.S. degree from Anyang University, February 2010. Her research interests include computer vision, visual tracking, and visual surveillance.

Gil-Jin Jang is an assistant professor at Ulsan National Institute of Science and Technology (UNIST), South Korea. He received his B.S., M.S., and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejon, South Korea in 1997, 1999, and 2004, respectively. From 2004 to 2006 he was a research staff at Samsung Advanced Institute of Technology and from 2006 to 2007 he worked as a research engineer at Softmax, Inc. in San Diego. From 2008 to 2009 he joined Hamilton Glaucoma center at University of California, San Diego as a postdoctoral employee. His research interests include acoustic signal processing, pattern recognition, speech recognition and enhancement, and biomedical signal engineering.

Bohyng Han received the BS and MS degrees from Seoul National University, Korea, in 1997 and 2000, respectively, and the PhD degree from the University of Maryland, College Park, in 2005. He was a senior research engineer at the Samsung Electronics R&D Center, Irvine, California and Mobileye Vision Technologies, Princeton, New Jersey. He is currently an assistant professor in the Department of Computer Science and Engineering at POSTECH, Korea. His research interests include statistical analysis in computer vision, machine learning, pattern recognition, and computer graphics.