# Visual Tracking Using Pertinent Patch Selection and Masking

Dae-Youn Lee[†], Jae-Young Sim[‡], and Chang-Su Kim[†]
[†]School of Electrical Engineering, Korea University, Seoul, Korea
[‡]School of Electrical and Computer Engineering,
Ulsan National Institute of Science and Technology, Ulsan, Korea
daeyounlee@mcl.korea.ac.kr, jysim@unist.ac.kr, changsukim@korea.ac.kr

## Abstract

*A novel visual tracking algorithm using patch-based appearance models is proposed in this paper. We first divide the bounding box of a target object into multiple patches and then select only pertinent patches, which occur repeatedly near the center of the bounding box, to construct the foreground appearance model. We also divide the input image into non-overlapping blocks, construct a background model at each block location, and integrate these background models for tracking. Using the appearance models, we obtain an accurate foreground probability map. Finally, we estimate the optimal object position by maximizing the likelihood, which is obtained by convolving the foreground probability map with the pertinence mask. Experimental results demonstrate that the proposed algorithm outperforms state-of-the-art tracking algorithms significantly in terms of center position errors and success rates.*

## 1. Introduction

Object tracking is a fundamental vision tool to facilitate various higher-level applications, including surveillance, object recognition, event analysis, and intelligent robotics. Even though many attempts have been made to develop efficient tracking algorithms, it is still challenging to detect and trace objects with illumination variations, pose changes, complex motions, and background clutters in a reliable manner. For robust tracking under these adverse conditions, it is essential to design effective appearance models.

Recently, many appearance models have been proposed [22, 24], and tracking algorithms can be roughly divided into two categories according to their appearance models: bounding box models [3, 4, 7, 9] and patch models [5, 11, 12, 14, 16]. A bounding box model uses the entire bounding box of a target object to extract object features, such as color, texture, and motion. It is sensitive to rapid and severe changes in structural appearance, which often occur in dynamic sequences, *e.g.* movies and sports videos. On the other hand, a patch model divides the bounding box into multiple smaller patches and extracts features for each patch separately. It can address appearance changes in a target object more flexibly, but it may decrease tracking accuracy when some foreground patches are not clearly distinguishable from background patches.

In this paper, we propose novel appearance models for both foreground and background to achieve reliable and accurate tracking. We first decompose the bounding box in the first frame into multiple patches and then select only pertinent patches, whose color histograms are frequently observed near the center of the bounding box, to construct the foreground appearance model. Moreover, we design multiple background appearance models to represent color histograms locally and adaptively. Then, by exploiting the foreground and background appearance models, we obtain the foreground probability map. Finally, we determine the optimal object position by convolving the foreground probability map with the pertinence mask, which records the likelihood that each pixel location within the bounding box belongs to the target object. This work has the following contributions:

1. Pertinent patch selection for an accurate foreground appearance model.
2. Localized multiple background appearance models.
3. Convolution scheme between the foreground probability map and the pertinence mask to suppress background clutters.

The rest of the paper is organized as follows: Section 2 summarizes related work. Section 3 overviews the proposed algorithm in the Bayesian framework. Section 4 proposes

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science, ICT & Future Planning (MSIP) (No. 2009-0083495), in part by the NRF grant funded by the Korea government (MSIP) (No. 2012-011031), and in part by Basic Science Research Program through the NRF funded by the Ministry of Education (2013R1A1A2011920).

the appearance models, and Section 5 describes the tracking process. Section 6 presents experimental results. Finally, Section 7 draws conclusions.

## 2. Related Work

**Histogram models:** Comaniciu *et al.* [7] proposed a nonrigid object tracking algorithm, which detects a target object to minimize the Bhattacharyya distance between the color histograms of reference and target bounding boxes. Their histogram-based appearance model, however, is sensitive to occlusions, since it loses spatial information. To alleviate this drawback, Adam *et al.* [1] divided the reference bounding box into multiple patches to extract patch-based histograms separately. He *et al.* [10] also decomposed a target object into overlapping regions, and constructed a histogram for each region using different weights of pixels.

**Local part tracking:** Hua and Wu [11] tracked local parts of a target object independently, and reduced the false detection rate using the relationships among the local parts. Nejhum *et al.* [19] approximated a target object with a small number of rectangular blocks, tracked, and refined the block positions based on the object contour. Kwon and Lee [14] employed a star model to connect local patches to the object center. Čehovin *et al.* [5] proposed a coupled-layer model, which combines local appearance with global appearance to describe a target object. They connected local patches using a triangulated mesh. Also, Yao *et al.* [23] proposed an online learning algorithm to exploit the relation between an entire object and its local patches implicitly.

**Patch-based appearance models:** Tang and Peng [20] employed patch models in two scales: large scale patches are used to discard unreliable small scale patches, and small scale patches are used to estimate the confidence of each input patch. In [16, 12], sparse dictionaries are used to describe patches in a bounding box. Liu *et al.* [16] measured the similarity between two objects, based on the sparse coefficient histograms of the patches within those objects. For more accurate tracking, Jia *et al.* [12] proposed an alignment pooling algorithm, which used sparse coefficients directly, instead of histograms or kernel densities, to measure the similarity. These algorithms, however, consider foreground appearance models only, and thus may fail when the background contains a region similar to the target object.

**Foreground probability (or confidence) map:** Avidan [3] proposed the ensemble classifier to estimate a foreground probability map, on which the mean shift localization is performed to detect a target. Wang *et al.* [21] segmented an image into superpixels, and estimated the foreground probabilities of the superpixels. They adopted the Bayesian tracking framework [2] to track a target based on the probability map.

## 3. Bayesian Tracking

We adopt the Bayesian framework [2] to formulate the proposed tracking algorithm. Let $\mathbf{x}_t$ and $\mathbf{z}_t$ be the state and the observation at time $t$, respectively. The posterior probability of $\mathbf{x}_t$ given the observations $\mathbf{z}_{1:t} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t\}$ can be written as

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) = \alpha_t p(\mathbf{z}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{z}_{1:t-1}), \tag{1}$$

where $\alpha_t$ is a normalization term, $p(\mathbf{z}_t|\mathbf{x}_t)$ is the likelihood, and $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$ is the prior probability of the state $\mathbf{x}_t$. We define the state $\mathbf{x}_t$ as the position of the bounding box $\Omega_t$ for the tracked object at time $t$. The proposed tracking algorithm finds the optimal $\hat{\mathbf{x}}_t$ to maximize the posterior probability of the position given the appearance information $\mathbf{z}_{1:t}$,

$$\hat{\mathbf{x}}_t = \underset{\mathbf{x}_t}{\operatorname{argmax}}\, p(\mathbf{x}_t|\mathbf{z}_{1:t}). \tag{2}$$

The prior probability of $\mathbf{x}_t$ is assumed to be uniformly distributed within a search region, given by

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \begin{cases} \frac{1}{N} & \text{if } \mathbf{x}_t \in R_t, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

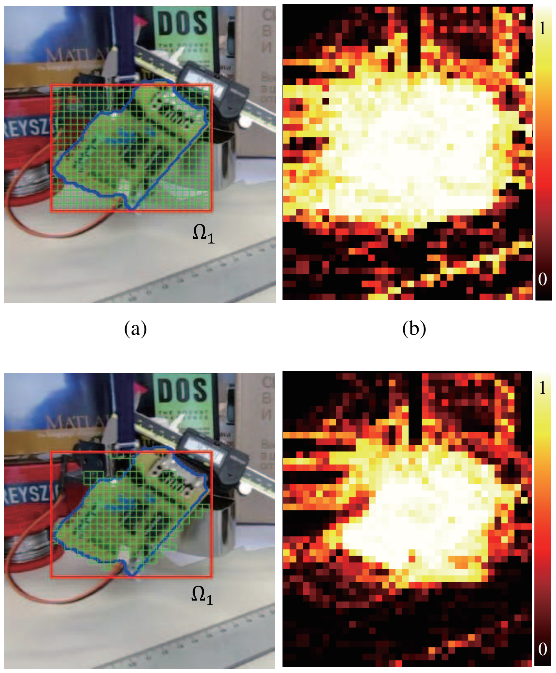
where $R_t$ is the search region, centered at the estimated position $\hat{\mathbf{x}}_{t-1}$ of the bounding box $\Omega_{t-1}$ at time $t-1$. Also, $N$ is the number of candidate positions in $R_t$. Hence, maximizing the posterior in (2) is equivalent to maximizing the likelihood $p(\mathbf{z}_t|\mathbf{x}_t)$. Thus, the likelihood design is one of the most important factors in object tracking.

## 4. Patch-Based Appearance Models

Our appearance models use an HSV color histogram with 48 bins: 16 bins for each channel of the hue, saturation, and value. However, extracting a color histogram from an entire bounding box may lose local color information, leading to inaccurate tracking [7]. We propose patch-based appearance models for the foreground and the background, respectively, which obtain a color histogram locally from each image patch smaller than the bounding box.

### 4.1. Foreground Appearance Model

In the first frame, the bounding box $\Omega_1$ of a foreground object is provided manually or by an object detection algorithm. Figure 1(a) illustrates the bounding box (red), which contains the object to be tracked (blue). We decompose the bounding box $\Omega_1$ into non-overlapping patches of size $8 \times 8$ and obtain a color histogram from each patch. In Figure 1(a), the bounding box contains background information as well, degrading the foreground probability map in Figure 1(b). To construct a more accurate foreground appearance model, we select only pertinent patches, which convey the foreground information, from the bounding box automatically.



(a) (b)

(c) (d)

Figure 1. Pertinent patch selection: All patches in the bounding box in (a) are used to obtain the foreground probability map in (b), whereas only pertinent patches in (c) are used to obtain the better map in (d). The bounding box is shown in red, the selected patches in green, and the object to be tracked in blue, respectively.

We observe that foreground patches tend to be near the center of the bounding box, whereas background patches near the box boundary. Moreover, even when a background patch is located near the center, it often has similar patches in terms of appearance near the boundary. Based on these observations, we assign a pertinence score to each patch. The notion of pertinence is related to the saliency based on the histogram contrast [6]. In [6], the saliency of a pixel is proportional to its rarity, which is defined as the sum of the differences from the pixel to the other pixels. On the contrary, the proposed pertinence score of a patch represents the frequency of the patch appearance within the bounding box.

We define a shrunken region $\Omega_1^S$ and an expanded region $\Omega_1^E$, which have the same center as the bounding box $\Omega_1$. When the size of $\Omega_1$ is $w \times h$, those of $\Omega_1^S$ and $\Omega_1^E$ are $0.6w \times 0.6h$ and $(w + 16) \times (h + 16)$, respectively. We decompose the expanded region $\Omega^E$ into non-overlapping patches of size $8 \times 8$ and obtain their color histograms. We then compute the saliency $s^E(i)$ of the $i$th patch in $\Omega_1$, with respect to the expanded region $\Omega_1^E$, as

$$s^E(i) = \min \left\{ \sum_{j=1}^{K} \|\mathbf{c}(i) - \mathbf{c}^E(j)\| \right\}, \tag{4}$$

where $\mathbf{c}(i)$ is the color histogram of the $i$th patch in $\Omega_1$, and $\mathbf{c}^E(j)$ is the color histogram of the $j$th selected patch from $\Omega_1^E$. Note that $s^E(i)$ minimizes the difference between $\mathbf{c}(i)$ and the selected $\mathbf{c}^E(j)$'s. Therefore, $s^E(i)$ is the sum of the distances from $\mathbf{c}(i)$ to its $K$ nearest neighbor histograms within the expanded region $\Omega_1^E$.

Similarly, we compute the saliency $s^S(i)$ of the $i$th patch in $\Omega_1$, with respect to the shrunken region $\Omega_1^S$,

$$s^S(i) = \min \left\{ \sum_{j=1}^{K} \|\mathbf{c}(i) - \mathbf{c}^S(j)\| \right\}, \tag{5}$$

where $\mathbf{c}^S(j)$ is the color histogram of the $j$th selected patch from $\Omega_1^S$. Note that the $i$th patch is likely to be a foreground one when $s^S(i)$ is small. This is because a foreground patch tends to have many similar patches within $\Omega_1^S$. In contrast, a background patch often has a large saliency $s^S(i)$.

Next, we compute the pertinence score $\psi(i)$ for the $i$th patch in $\Omega_1$ as

$$\psi(i) = \frac{s^E(i)}{s^S(i)}. \tag{6}$$

Note that $0 \leq \psi(i) \leq 1$, since $\Omega_1^S \subset \Omega_1^E$ and thus $s^S(i) \geq s^E(i)$. When the $i$th patch contains the background information, $s^E(i)$ is usually much smaller than $s^S(i)$. In general, a background patch has similar patches in the expanded region, but not in the shrunken region. In contrast, when the $i$th patch contains the foreground information, $s^E(i)$ and $s^S(i)$ tend to be similar, and the pertinence score $\psi(i)$ is near 1. Therefore, the pertinence score $\psi(i)$ indicates the likelihood that the $i$th patch belongs to the foreground object.

If $K$ in (4) or (5) equals the number of all patches in $\Omega_1^E$ or $\Omega_1^S$, $s^E(i)$ or $s^S(i)$ becomes the histogram contrast [6]. However, when $K$ becomes large, some small regions within the foreground object may yield larger $s^S$ and small pertinence scores. Therefore, we fix $K = 4$.

We select the $i$th patch as pertinent patch, when $\psi(i) > \gamma$. We set $\gamma = 0.56$. To remove outliers from the pertinent patch selection, we group the selected patches into connected components. Then, we eliminate the connected components whose sizes are smaller than the quarter of the largest component. In Figure 1(c), green patches represent pertinent ones. The pertinent patch selection improves the accuracy of the foreground probability map, as shown in Figure 1(d).

### 4.2. Multiple Background Models

The patch-based model has the advantages in handling photometric and geometric changes in a target object, but also the weakness that background patches are less distinguishable from foreground patches with a smaller patch size. The conventional bounding box models [3, 9] construct a single background model. However, in the proposed

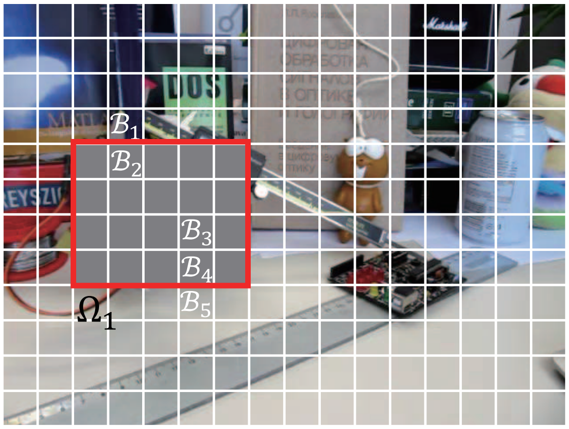


Figure 2. Background model construction: For blocks within the bounding box $\Omega_1$, the background models are imported from the nearest available blocks, *i.e.* $\mathcal{B}_2$ from $\mathcal{B}_1$, and $\mathcal{B}_3$ and $\mathcal{B}_4$ from $\mathcal{B}_5$.

patch-based approach, the single background model does not provide sufficient information for the tracker to separate background patches from foreground ones. Therefore, we propose using multiple background models.

We divide an input image into non-overlapping blocks of size $8 \times 8$ and construct a background model at each block location. Let $\mathcal{B}_j$ denote the background model for the $j$th block location, which maintains recent color histograms observed at that location. In the first frame, $\mathcal{B}_j$ is initialized with the color histogram of the $j$th block, if it is outside the bounding box $\Omega_1$. On the other hand, if the $j$th block is within $\Omega_1$, $\mathcal{B}_j$ is initialized with the color histogram of the nearest available block. For example, in Figure 2(a), the background models $\mathcal{B}_2$, $\mathcal{B}_3$, and $\mathcal{B}_4$ import the background color histograms from the nearest available blocks outside $\Omega_1$: $\mathcal{B}_2$ from $\mathcal{B}_1$, and $\mathcal{B}_3$ and $\mathcal{B}_4$ from $\mathcal{B}_5$.

From the second frame, we update the background models according to tracking results. After estimating the location of the bounding box $\Omega_t$, we update the models only for the blocks outside $\Omega_t$. We add the color histograms of those blocks into the corresponding background models, which are implemented as queues. Each queue keeps ten histograms, and the oldest histogram is discarded when a new histogram is added.

## 5. Tracking

At time $t$, we define a search region $R_t$, which includes the bounding box $\Omega_{t-1}$ obtained at time $t-1$. Then, we estimate the foreground probability of each pixel within $R_t$, by employing the foreground and background models. We estimate the likelihood $p(\mathbf{z}_t|\mathbf{x}_t)$ in (1), by convolving the foreground probability map with the pertinence mask. Finally, we obtain the optimal position $\hat{\mathbf{x}}_t$ of the bounding box $\Omega_t$, which maximizes the likelihood $p(\mathbf{z}_t|\hat{\mathbf{x}}_t)$

### 5.1. Foreground Probability

Suppose that the size of $\Omega_{t-1}$ is $w \times h$. Then, we set the size of the search region $R_t$ to $(w+2\delta) \times (h+2\delta)$, where $\delta$ is set to 30 in this work. We divide $R_t$ into non-overlapping patches of size $8 \times 8$. We extract the input color histogram $\mathbf{c}_{\text{in}}^m$ from the $m$th patch $\mathcal{P}_m$. Among the foreground histograms of the pertinent patches, we find the two nearest histograms $\mathbf{c}_{\text{f}}^1$ and $\mathbf{c}_{\text{f}}^2$ to $\mathbf{c}_{\text{in}}^m$ in the feature space, using the randomized $k$-d trees [17]. We also employ the cross-bin metric [15] to measure the distance between histograms to reduce the effects of quantization errors in the histogram construction. Then, we compute the foreground distance $d_{\text{f}}$ as the average of the two nearest distances.

To compute the background distance $d_{\text{b}}$, we employ only the 25 local background models, which are geometrically close to the $m$th patch $\mathcal{P}_m$, instead of using all background models. Similarly to $d_{\text{f}}$, we compute the background distance $d_{\text{b}}$ as the average distance from the input histogram $\mathbf{c}_{\text{in}}^m$ to the two nearest histograms $\mathbf{c}_{\text{b}}^1$ and $\mathbf{c}_{\text{b}}^2$ in the 25 background models.

Consequently, the foreground probability of each pixel $\mathbf{u}$ in $\mathcal{P}_m$ is given by

$$\Gamma(\mathbf{u}) = \left( \frac{d_b}{d_f + d_b} \right)^2. \tag{7}$$

We normalize $\Gamma(\mathbf{u})$ into the range of $[0, 1]$, and then set $\Gamma(\mathbf{u}) = 0$ when the normalized value is smaller than $0.9$.

### 5.2. Pertinence Masking for Likelihood Estimation

We may select the position $\mathbf{x}_t$, which maximizes the sum of the foreground probabilities within the corresponding bounding box, as in [3]. This approach is effective, when the bounding box includes only foreground pixels, as shown in Figure 3(a). However, when the bounding box includes some background pixels with relatively large foreground probabilities, as shown in Figure 3(b), it may yield an inaccurate result. Therefore, we suppress the foreground probabilities of those background pixels using the pertinence mask $\mathcal{M}_t$ in Figure 3(c).

The pertinence mask $\mathcal{M}_t$ is defined as the window of foreground probabilities, which is updated at time $t-1$. It has the same size as the bounding box. Then, we compute the likelihood $p(\mathbf{z}_t|\mathbf{x}_t)$ by

$$\begin{aligned} & p(\mathbf{z}_t|\mathbf{x}_t) \\ & = \frac{1}{|\mathcal{M}_t|} \sum_{\mathbf{k}} \big( \Gamma(\mathbf{x}_t + \mathbf{k}) \mathcal{M}_t(\mathbf{k}) + \bar{\Gamma}(\mathbf{x}_t + \mathbf{k}) \bar{\mathcal{M}}_t(\mathbf{k}) \big), \end{aligned} \tag{8}$$

where $|\mathcal{M}_t|$ is the number of pixels within the mask, $\mathbf{k}$ denotes the relative position in $\mathcal{M}_t$, $\bar{\Gamma}(\cdot) = 1 - \Gamma(\cdot)$, and $\bar{\mathcal{M}}(\cdot) = 1 - \mathcal{M}(\cdot)$. The first term $\Gamma(\mathbf{x}_t + \mathbf{k})\mathcal{M}_t(\mathbf{k})$ in (8)

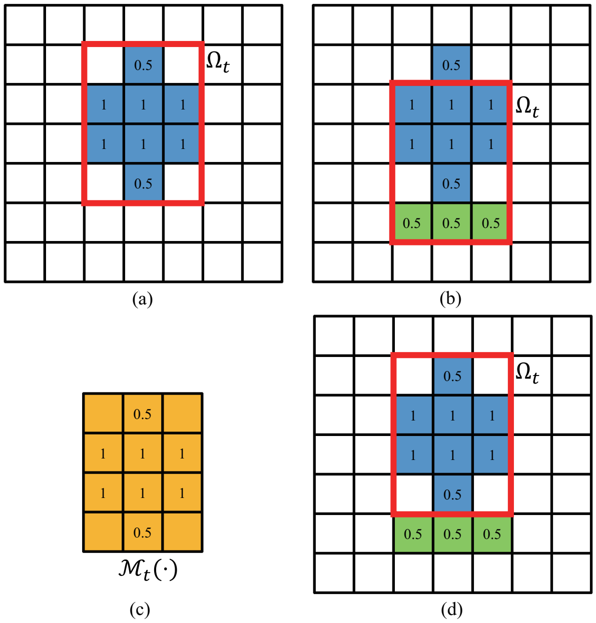


Figure 3. Likelihood estimation using the pertinence mask: Foreground and background pixels are shown in blue and green, respectively. The numbers are the foreground probabilities. The bounding box is at the correct position in (a), whereas the bounding box is drifted incorrectly due to the background pixels with high probabilities in (b). The pertinence mask $\mathcal{M}_t$ in (c) prevents this drift in (d).

counts valid foreground probabilities, matched with the pertinence mask, while the second term $\bar{\Gamma}(\mathbf{x}_t + \mathbf{k})\bar{\mathcal{M}}_t(\mathbf{k})$ implicitly gives penalties to foreground probabilities on non-pertinent pixel locations. Using the pertinence mask, we can track the object more reliably, as shown in Figure 3(d).

The foreground probabilities of pixels in a target object vary at each frame. When the tracked position $\hat{\mathbf{x}}_t$ satisfies $p(\mathbf{z}_t|\hat{\mathbf{x}}_t) > 0.75$, we update the pertinence mask $\mathcal{M}_t$ to $\mathcal{M}_{t+1}$ via

$$\mathcal{M}_{t+1}(\mathbf{k}) = (1-\lambda)\mathcal{M}_t(\mathbf{k}) + \lambda\Gamma(\hat{\mathbf{x}}_t + \mathbf{k}) \tag{9}$$

where $\lambda$ is an update factor. It is fixed to $\lambda = 0.0005$. If the tracked position does not satisfy the condition, $\mathcal{M}_{t+1} = \mathcal{M}_t$.

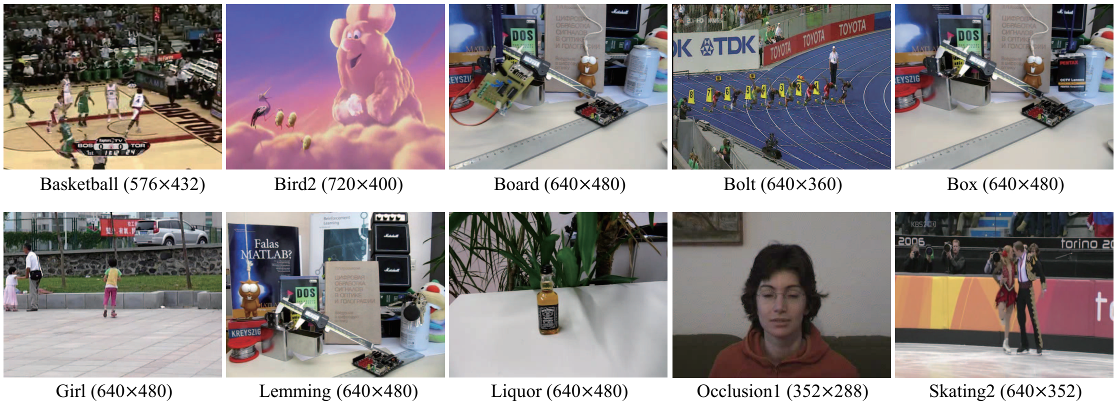
## 6. Experimental Results

We report the performance of the proposed algorithm on ten test sequences: “Liquor,” “Box,” “Board,” “Lemming,” [18] “Basketball,” “Skating2,” [13] “Bolt,” “Bird2,” “Girl,” [21] and “Occlusion1” [1], whose sample frames are shown in Figure 4. We compare the proposed algorithm with four state-of-the-art trackers: STRUCK tracker (ST) [9], superpixel tracker (SPT) [21], compressive tracker (CT) [25], and local histogram tracker (LHT) [10]. We use illumination invariant features for LHT, since these features yield better tracking results than intensity features. The proposed algorithm is implemented in C++ without optimization, and achieves the average processing speed of 3.4 frames per second on a computer with a 3.3 GHz processor and 8 Gbyte RAM.

Figure 5 compares the tracking accuracy of the proposed algorithm with those of the conventional algorithms, in terms of center position errors. A center position error is defined as $\|\mathbf{x}_g - \hat{\mathbf{x}}\|$, where $\mathbf{x}_g$ is the center position of the ground truth bounding box and $\hat{\mathbf{x}}$ is its estimated position by a tracker. The proposed algorithm provides smaller center position errors than the conventional algorithms on most test sequences, especially on “Basketball,” “Bird2,” “Bolt,” “Box,” “Lemming,” and “Skating2.” In “Basketball,” “Bolt,” and “Skating2,” there are fast object motions, but the proposed algorithm handles the rapid structural changes effectively using patch-based foreground appearance models. Moreover, in “Basketball,” “Bolt,” and “Box,” the proposed algorithm alleviates the effects of background clutters by employing multiple background models. In “Board,” “Bird2,” and “Lemming,” the proposed algorithm suppresses background information within initial bounding boxes, based on the pertinent patch selection and masking, to provide reliable tracking results. Note that the proposed algorithm yields relatively bad performance on the beginning part of the “Girl” sequence. It is because the girl is fully occluded by a man at the 111th frame.

Table 1 compares the average center position errors, as well as the average success rates that are measured by the PASCAL scoring method [8]. The PASCAL method declares a frame as successful, when the overlapping area between the ground truth bounding box and the estimated bonding box is larger than half of the total area occupied by the two boxes. The PASCAL method then counts the number of successful frames. For each metric on each test sequence, the best performance and the second best one are marked in bold fonts and underlined, respectively. We observe that the proposed algorithm yields the center position error of 20 pixels and the success rate of 85% on average, which outperforms the conventional algorithms significantly.

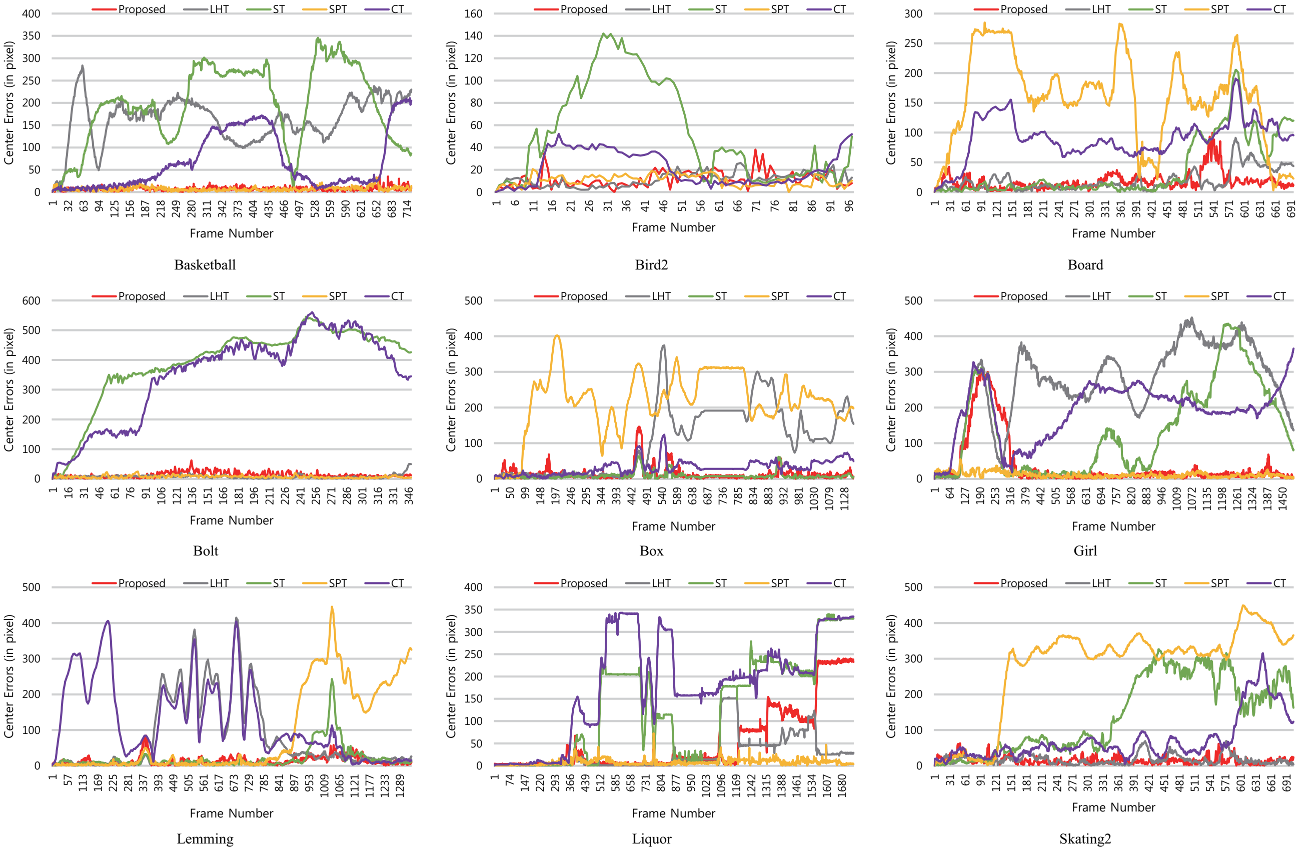
Figure 6 compares the tracking results qualitatively. In the “Board” and “Skating2” sequences, the initial bounding boxes include large portions of the background. Therefore, the conventional algorithms cannot track the target objects correctly. In contrast, the propose algorithm constructs an accurate foreground appearance model by excluding those background patches, and tracks the objects reliably. Also, some small parts in the background, which are similar to a target object, degrade the tracking performance of the conventional algorithms. For example, in “Basketball,” the players wear the uniform with the same green color. Hence, the conventional algorithms suffer from the ambiguity. On



Basketball (576×432) Bird2 (720×400) Board (640×480) Bolt (640×360) Box (640×480)

Girl (640×480) Lemming (640×480) Liquor (640×480) Occlusion1 (352×288) Skating2 (640×352)

Figure 4. Test video sequences.



Basketball Bird2 Board

Bolt Box Girl

Lemming Liquor Skating2

Figure 5. Comparison of the center position errors of the proposed algorithm and the conventional algorithms: ST [9], SPT [21], CT [25], and LHT [10].

the contrary, the proposed algorithm employs only local information to model the background appearance. Thus, we can alleviate the effects of such ambiguity.

## 7. Conclusions

In this paper, we proposed a robust visual tracking algorithm, which uses patch-based appearance models adaptively. We first introduced the notion of pertinence score to construct a more accurate foreground model by excluding the background information within a bounding box. We also proposed using multiple background models to represent different locations locally and adaptively. We generated a foreground probability map, which was then convolved with the pertinence mask to suppress the effects of background clutters. Experimental results demonstrated

Table 1. Comparison of the center position errors (CE) and the success rates (SR) [8] between the proposed algorithm and the conventional algorithms: ST [9], SPT [21], CT [25], and LHT [10]. The best result is marked in bold fonts and the second best result is underlined.

| Sequence | CE | | | | | SR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ST | SPT | CT | LHT | Proposed | ST | SPT | CT | LHT | Proposed |
| Basketball | 195 | **7** | 67 | 163 | **7** | 0.03 | 0.83 | 0.26 | 0.02 | **0.98** |
| Bird2 | 54 | **11** | 22 | 12 | 12 | 0.36 | **0.97** | 0.53 | 0.93 | 0.9 |
| Board | 38 | 158 | 90 | 19 | **18** | 0.7 | 0.14 | 0.09 | 0.94 | **0.95** |
| Bolt | 392 | **7** | 352 | 8 | 13 | 0.02 | 0.67 | 0 | **0.78** | 0.56 |
| Box | **9** | 217 | 32 | 108 | 14 | **0.95** | 0.08 | 0.39 | 0.4 | 0.9 |
| Girl | 138 | **12** | 191 | 269 | 38 | 0.2 | **0.95** | 0.04 | 0.07 | 0.79 |
| Lemming | 21 | 89 | 125 | 83 | **13** | 0.8 | 0.59 | 0.19 | 0.47 | **0.88** |
| Liquor | 128 | **8** | 179 | 27 | 53 | 0.4 | **0.99** | 0.21 | 0.72 | 0.67 |
| Skating2 | 142 | 278 | 73 | 17 | **15** | 0.19 | 0.03 | 0.16 | 0.7 | **0.83** |
| Occlusion1 | 17 | 34 | 20 | **13** | 17 | **1** | 0.26 | 0.98 | **1** | 0.99 |
| Average | 113 | 82 | 115 | 72 | **20** | 0.47 | 0.55 | 0.29 | 0.6 | **0.85** |

that the proposed algorithm achieves more accurate tracking results than the conventional state-of-the-art trackers.

## References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, pages 798–805, 2006. 2, 5

[2] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *TSP*, 50(2):174–188, 2002. 2

[3] S. Avidan. Emsemble tracking. *TPAMI*, 29(2):261–271, Feb 2007. 1, 2, 3, 4

[4] C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust L1 tracker using accelerated proximal gradient approach. In *CVPR*, pages 1830–1837, 2012. 1

[5] L. Čehovin, M. Kristan, and A. Leonardis. Robust visual tracking using an adaptive coupled-layer visual model. *TPAMI*, 35(4):941–953, Apr. 2013. 1, 2

[6] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu. Global contrast based salient region detection. In *CVPR*, pages 409–416, 2011. 3

[7] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *TPAMI*, 25(5):564–575, May 2003. 1, 2

[8] M. Everingham, L. Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010. 5, 7

[9] S. Hare, A. Saffari, and P. Torr. Struck: Structured output tracking with kernels. In *ICCV*, pages 263–270, 2011. 1, 3, 5, 6, 7, 8

[10] S. He, Q. Yang, R. W. Lau, J. Wang, and M.-H. Yang. Visual tracking via locality sensitive histograms. In *CVPR*, 2013. 2, 5, 6, 7, 8

[11] G. Hua and Y. Wu. Measurement integration under inconsistency for robust tracking. In *CVPR*, volume 1, pages 650–657, 2006. 1, 2

[12] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*, pages 1822–1829, 2012. 1, 2

[13] J. Kwon and K. M. Lee. Visual tracking decomposition. In *CVPR*, pages 1269–1276, 2010. 5

[14] J. Kwon and K. M. Lee. Highly nonrigid object tracking via patch-based dynamic appearance modeling. *TPAMI*, 35(10):2427–2441, Oct. 2013. 1, 2

[15] I. Leichter. Mean shift trackers with cross-bin metrics. *TPAMI*, 34(4):695–706, Feb 2012. 4

[16] B. Liu, J. Huang, L. Yang, and C. Kulikowski. Robust tracking using local sparse appearance model and k-selection. In *CVPR*, 2011. 1, 2

[17] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Int'l Conf. Computer Vision Theory Appl. (VISSAPP)*, pages 331–340, 2009. 4

[18] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. PROST: Parallel robust online simple tracking. In *CVPR*, 2010. 5

[19] S. Shahed Nejhum, J. Ho, and M.-H. Yang. Visual tracking with histograms and articulating blocks. In *CVPR*, pages 1–8, 2008. 2

[20] M. Tang and X. Peng. Robust tracking with discriminative ranking lists. *TIP*, 21(7):3273–3281, 2012. 2

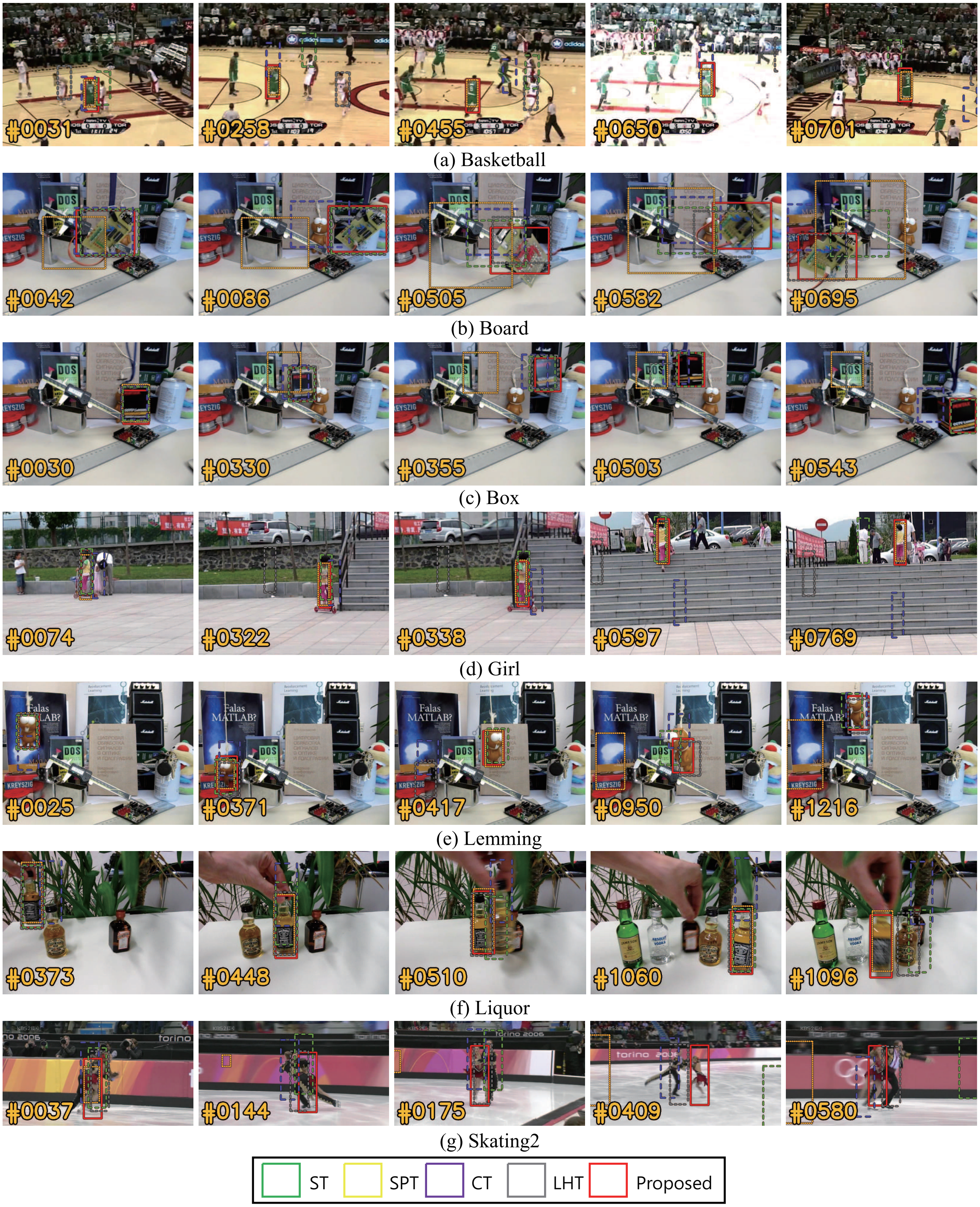
[21] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *ICCV*, pages 1323–1330, 2011. 2, 5, 6, 7, 8

[22] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823 – 3831, 2011. 1

[23] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel. Part-based visual tracking with online latent structural learning. In *CVPR*, pages 2363–2370, 2013. 2

[24] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), Dec. 2006. 1

[25] K. Zhang, L. Zhang, and M.-H. Yang. Real-time compressive tracking. In *ECCV*, pages 864–877, 2012. 5, 6, 7, 8



(a) Basketball

(b) Board

(c) Box

(d) Girl

(e) Lemming

(f) Liquor

(g) Skating2

Figure 6. Examples of the tracking results of the proposed algorithm and the conventional algorithms: ST [9], SPT [21], CT [25], and LHT [10].