# Adaptive Feature Extraction with Haar-like Features for Visual Tracking

Seunghoon Park
Adviser : Bohyung Han
Pohang University of Science and Technology
Department of Computer Science and Engineering
pclove1@postech.ac.kr

December 26, 2010

**Abstract**

We propose an adaptive feature extraction technique to improve the performance of visual tracking algorithm. The goal is achieved by the integration of Haar-like features in *ensemble tracking* [1] framework. Our experiment shows that the proposed algorithm performs well even in case of various illumination changes and short occlusions.

## 1 Introduction

Visual tracking is one of the most well-known problems in computer vision because it has a variety of applications such as surveillance systems [2], medical image analysis [3], human-computer interaction [4], and so on. The goal of visual tracking is to identify the state of target objects throughout the sequence. A visual tracker is typically composed of several components; the feature description of target, the appearance model of target based on the features, and the controller handling the dynamics of target. Although many approaches have been introduced so far, visual tracking is still an challenging problem due to various difficult situations such as appearance changes, illumination variations and occlusions.

### 1.1 Related Work and Our Approach

#### 1.1.1 Features

To represent target objects in visual tracking problems, various approaches have been utilized such as color information [5, 6, 7], texture [8], edge orientation [7], etc. The most frequently used feature is probably color histogram [5, 6, 7] because color is directly available from image, and histogram is fast to compute. However, there are some drawbacks of color histogram. First of all, color is very sensitive to illumination, so it is not appropriate for the situations where lighting changes frequently. Second, color histogram is not straightforward to handle spatial information of target, and not sufficiently discriminative sometimes. So, we integrate Haar-like features instead of color information, and the properties of the Haar-like feature are described below.

**Haar-like Features** As an attempt to develop a different feature to describe target objects, Papageorgiou et al. have introduced a general framework for object detection using a Haar wavelet representation [9]. Motivated by the work by Papageorgiou et al., Viola and Jones

have proposed a face detection algorithm based on Haar-like features, and new image representation for extracting Haar-like features efficiently, which is called *integral image* (aka summed area table) [10]. Fast computation is an important advantage of integral image and only several accesses to the integral image are required to extract a Haar-like feature response. Also, Haar-like features are more robust to illumination changes than color histogram. Lienhart and Maydt also provided an extended set of Haar-like features [11]. Speeded Up Robust Features (SURF) proposed by Bay et al. [12] is a good practical example of Haar-like features. According to the authors, SURF provides reasonable repeatability, distinctiveness, and robustness. In addition, the speed of computations is relatively fast by incorporating integral image in their implementation.

### 1.1.2 Feature Extraction

The methods of feature extraction can be divided to two categories based on whether features are adaptive to changes of the target appearance or not. The adaptive feature extraction technique switches one feature to another among pre-defined feature set, or generates a new feature in each time frame based on observations, while the non-adaptive method keeps the same features given at the first frame. An example of the adaptive method is Collins et al.'s [13]. They used the linear combinations of RGB values as the candidate feature set and ranked them in order of the discriminativeness between the target and the background. Then, the most discriminative $N$ features are selected at each frame and the estimation of target state is obtained from mean-shift processes. The authors also introduced the idea of peak difference to reduce the possibility of model drift caused by distracting objects near the target.

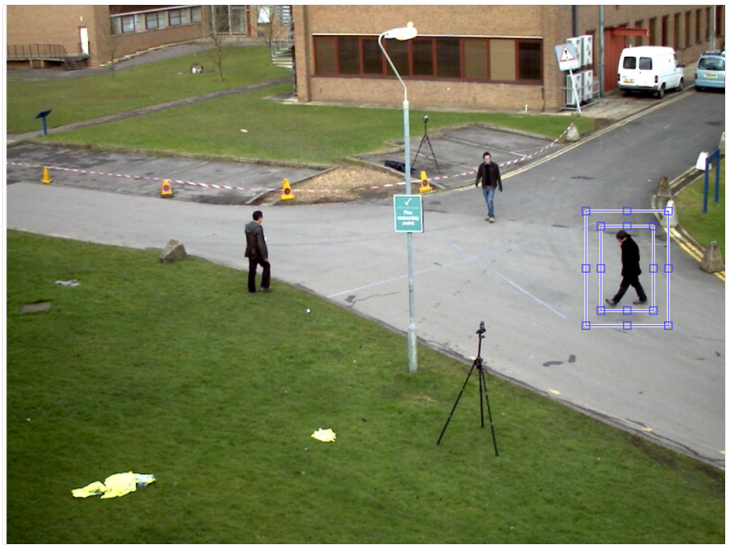
**Ensemble Tracking** Ensemble tracking originally proposed by Avidan [1] can be seen as a more improved version of on-line feature selection algorithm compared to [13] in that weights of the different features are assigned automatically. Also, while the technique proposed in [13] is limited to the use of histogram, ensemble tracking can work with any kind of high-dimensional features. Ensemble tracking used Histogram of Oriented Gradients(HOG) and RGB values as the candidates of features. In our work, we integrate Haar-like features instead of HOG and attempt to improve the performance of ensemble tracking algorithm.

The key concept of the approach proposed in this work is to integrate Haar-like features in ensemble tracking. The approach has impacts to two challenging problems in visual tracking: illumination variations and occlusions. Most color-based approaches suffer when lighting changes too frequently but the proposed algorithm is robust to illumination changes because the use of Haar-like features and the on-line weighting of each feature in every frame handle various challenging situations effectively. Moreover, the ensemble tracker [1] is inherently robust to occlusions, and the property is preserved with Haar-like features. Also, the use of the integral image makes the computation of feature values much faster.

The rest of the paper is organized as follows. Section 2 describes the overview of our algorithm in a foraml manner. Then, the details of Haar-like features and ensemble tracking are presented in Section 3 and 4, respectively. Section 5 illustrates the results of experiments on a few of datasets.

## 2 Problem Description

This research views visual tracking as a classification problem where there are two different regions: target area and background area and this approach was used already in other works[13, 1]. These regions are represented by two rectangles in a frame. The target box is inside of the background box. The pixels within the target box are labeled to target(or $+1$), while the pixels in the border area between two boxes are considered as background(or $-1$). Then, the goal of visual tracker is to find the proper location of two boxes that classify pixels very well to two regions over a frame. Figure 1 shows two target and background boxes, and the labels of sample pixels for each box; red points indicate target and blue point indicate background.



(a) Entire image

(b) Samples from foreground (red dots) and background (blue dots)

(c) Confidence Map

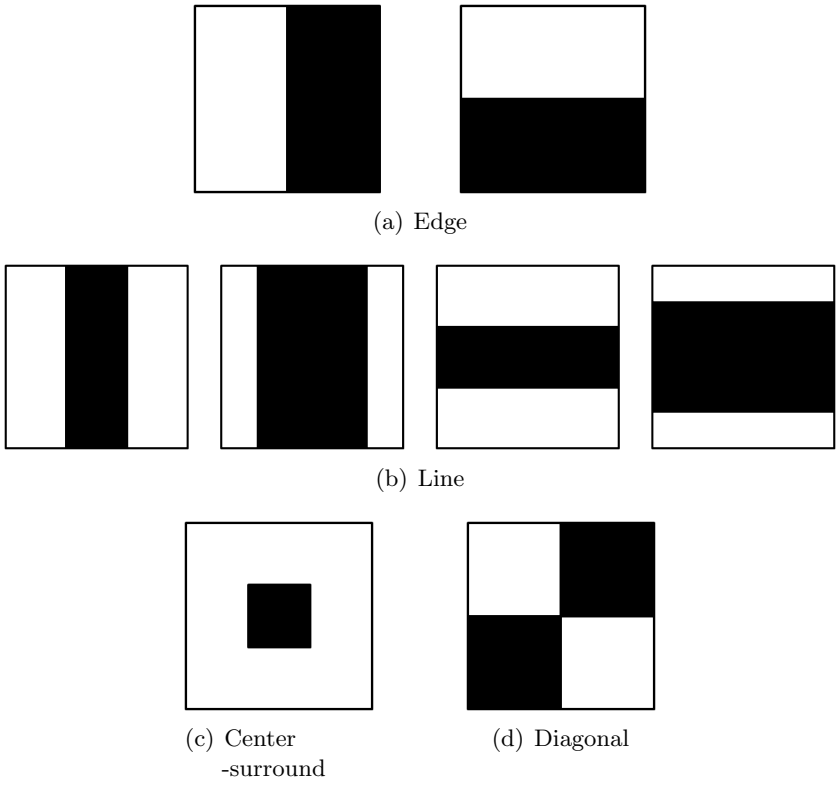
Figure 1: The overview of foreground/background classification technique with PETS 2009 workshop benchmark dataset [14].

In a more formal way, we get a d-dimensional feature vector $\mathbf{x} \in \mathbb{R}^d$ at each pixel. The components of the vector can be any real number such as RGB values and the outputs after applying a certain filter at the pixel etc. Then, we label each vector $\mathbf{x}$ to one of two classes

$\{+1, -1\}$ in the first frame, where $+1$ means target and $-1$ means background. After applying the vectors in the target box and the background box in a following frame to a classifier, the confidence value of each pixel can be achieved. The range of the confidence value is modified to [0, 1] by clipping negative values and rescaling, where the higher value means the higher possibility that the pixel belongs to target box. The Figure 1(c) shows a confidence map. The most convincing location at that frame is attained through a method like mean-shift process. The new labeling is performed based on this new location of two boxes and the whole process is repeated at each frame.

# 3 A Set of Haar-like Features

## 3.1 Prototype



(a) Edge

(b) Line

(c) Center -surround

(d) Diagonal

Figure 2: 8 prototypes of Haar-like features. We apply three different sizes of such filters ($5 \times 5$, $11 \times 11$, $21 \times 21$), which generate $24 = 8 \times 3$ Haar-like features.

A Haar-like feature is computed as subtracting the sums of pixel values in black rectangles from the sums of pixel values in white rectangles. As you can notice from the word 'pixel values', a color frame should be converted to a gray-scaled one. Although there are various ways to do it, weighting each color value(e.g. RGB) differently, the gray-scaled value is calculated simply as the follow:

$$\text{Gray-scaled Intensity} = 0.2989 \times R + 0.5870 \times G + 0.1140 \times B$$

In addition, a set of Haar-like features consists of $p$ prototypes with $s$ different sizes. Each prototype can capture a certain spatial information such as edge, line, surrounded pixels and

diagonal. This enables a visual tracker to deal with spatial information from Haar-like feature as well as color information from RGB values. Since the size of Haar-like feature affects the meaning captured in the Haar-like feature, $s$ different sizes are necessary to maximize the diversity. Figure 2 shows the eight prototypes used in this research.

### 3.2 Integral Image

Integral image initially introduced by Viola and Jones[10] is used to compute the Haar-like feature faster. The value of the integral image at a point is defined as the sum of all the pixels to the left and above. Namely, the integral image value at $(x, y)$ is defined as the follow:

$$\text{IntegralImage}(x, y) = \sum_{x' \leq x, y' \leq y} \text{pixel}(x', y')$$

Once the integral image is achieved, the sum of pixels within any rectangle is computed through only four array accesses to the integral image. The sum of pixels within a rectangle $R$ defined as $(min_x, min_y, w, h)$ can be achieved as the follow:

$$\begin{aligned} \text{SUM}(R) \quad = \quad & \text{IntegralImage}(min_x + w - 1, min_y + h - 1) + \text{IntegralImage}(min_x - 1, min_y - 1) \\ & - \text{IntegralImage}(min_x + w - 1, min_y - 1) - \text{IntegralImage}(min_x - 1, min_y + h - 1) \end{aligned}$$

where $(min_x, min_y)$ is the upper-left corner of the rectangle $R$, $w$ is its width and $h$ is its height. Figure 3 shows an example of the computation.

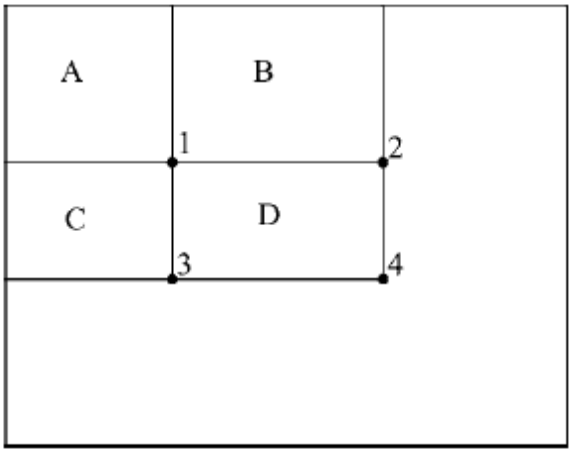



Figure 3: Integral image for Haar-like features. Note that the sum of the feature values within any rectangle is computed through 4 memory accesses (e.g.$D = (4 + 1) - (2 + 3)$) [10].

## 4 Ensemble Tracking

### 4.1 The Weak Classifier

Ensemble tracking is a kind of AdaBoost, so it consists of a few of weak classifiers. Before training a weak classifier, we need to get $N$ samples that are composed of $d$-dimensional feature vectors $\mathbf{x}_i \in \mathbb{R}^d$ and $N$ labels $y_i \in \{+1, -1\}$ for each feature vector, and their weights $w_i$ where

$i = 1, 2, \ldots, N$. Each weak classifier takes a feature vector $\mathbf{x}_i$ as an input and determines its class $\in\{+1, -1\}$. The weak classifier is defined as

$$h(\mathbf{x}) = sign(\mathbf{h}^T\mathbf{x})$$

where $\mathbf{h}$ is a hyperplane computed using weighted least square regression as following.

$$\mathbf{h} = (\mathbf{A}^T\mathbf{W}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}^T\mathbf{W}\mathbf{y}$$

where $\mathbf{A}$ is a $N \times (d+1)$ matrix whose each row is $\mathbf{x}_i$ augmented with the constant 1, $[\mathbf{x}_i, 1]$, $\mathbf{W}$ is a $N \times N$ diagonal matrix whose diagonal element is $w_i$.

T weak classifiers explained above constitute a strong classifier $H(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$. Whenever training a weak classifier $h_t(\mathbf{x})$, $t = 1, 2, \ldots, T$, its weight is computed as

$$\alpha_t = \frac{1}{2} log \frac{1 - err}{err}$$

where

$$err = \sum_{i=1}^{N} w_i |h_t(\mathbf{x}_i) - y_i|$$

In addition, the weights of samples are updated as the following.

$$w_i = w_i e^{(\alpha_t |h_t(\mathbf{x}_i) - y_i|)}$$

It should be stressed from the above equation that later classifiers focus more on a difficult sample to be classified.

### 4.2 Mean-shift process

After training T weak classifiers, we obtain new N samples in the next frame and use the strong classifier to get a confidence map $L$, where $L(r, c) = H(\mathbf{x}(r, c))$ and $(r, c) \in$ background box. If the previous location of target is represented as $(r, c)$, then the amount of change of $r$ in an iteration is

$$\Delta r = \frac{\sum_{(r_i, c_i) \in L} L(r_i, c_i) \times (r_i - r)}{\sum_{(r_i, c_i) \in L} L(r_i, c_i)}$$

and $\Delta c$ is computed similarly. This computation is repeated until the predicted location of target converges to a location. Then, new N samples are re-labeled based on the new location of target and background boxes.

### 4.3 Update of Ensemble Tracking

Once we determined the new location of target, we need to create new weak classifiers to adapt to the new frame. However, we remain $K (< T)$ weak classifiers to be robust to an occlusion. For example, if we delete and create $T$ weak classifiers in case of an occlusion there is a danger that they recognize an obstacle as the target. The $K$ weak classifiers are chosen in the following way. First, we re-train all $T$ weak classifiers based on the new $N$ samples and their new labels using the same way at the first frame. A weak classifier which has the minimal $err$ is chosen and its weight is updated in each iteration. This is repeated until $K$ weak classifiers are selected. After that, we make new $T - K$ weak classifiers, get a new strong classifier, and are ready to move on to the next frame.

# 5 Experiments

## 5.1 Implementation Issues

The experiments were performed in MATLAB and there are many parameters determined manually. Those are as the followings.

- `The number of protypes(`$p$`)` - 8 is used as shown Figure 2.
- `The number of different sizes of the protype(`$s$`)` - There are 3 different sizes of each prototype($5 \times 5$, $11 \times 11$ and $21 \times 21$).
- `The number of weak classifiers(`$T$`)` - 5 is used as same as the original ensemble tracking work.
- `The number of remained weak classifiers(`$K$`)` - 3 is used as same as the original ensemble tracking work.
- `The ratio of sizes of target and background boxes` - the size of background box is twice as the size of target box.
- `The number of samples at each frame(`$N$`)` - $N$ is a half of the number of pixels within target box and $\frac{N}{2}$ samples come from target and background region respectively.
- `The dimension of a feature vector(`$d$`)` - $d = p * s + 3$, 27(24 Haar-like features and RGB) in this experiment.

## 5.2 Results

The visual tracker proposed in this paper was tested on a few of sequences from PETS 2009 workshop benchmark data[14]. Figure 4 shows a successful tracking in case of a short occlusion. The test is performed from frame 138 to frame 163 and the tracked women is occluded by two men during three frames(144, 145 and 146). Look at the confidence map at frame 162 carefully how the tracker keeps appropriate classifiers for the women.

Figure 5 shows another traffic dataset from H.-H. Nagel in Universität Karlsruhe(TH)[15]. A white car entering to the shadow region from bright region was tracked from frame 136 to frame 270. It should be noticed that there is a huge difference of illumination between two regions and the tracked car enters to the shadow region twice(at 161 and 219). Because the dataset consisted of gray-scaled images, only 24 Haar-like features were used.

### 5.2.1 Limitations

**Long Occlusions** Although the tracker can deal with partial occlusions over a few of frames, it gets stuck with a near obstacle when occlusions occur over many frames. A walking person is being tracked in Figure 6 but the tracker gets stuck with the sign after nine frames he was occluded by the sign.

**Edge Phenomena** The confidence maps sometimes don't look like the target but the edge of the target. An example of this phenomenon is shown in Figure 7. The most ideal confidence map is an image, where the only pixels within the target have values larger than 0 and the values become larger as the points are getting close to the center of the target. However, these phenomena don't affect the final results of locating the target because mean-shift processes still can generate proper outcomes using edges of the target.

(a) Frame 139 (b) Frame 144

(c) Frame 145 (d) Frame 146

(e) Frame 162

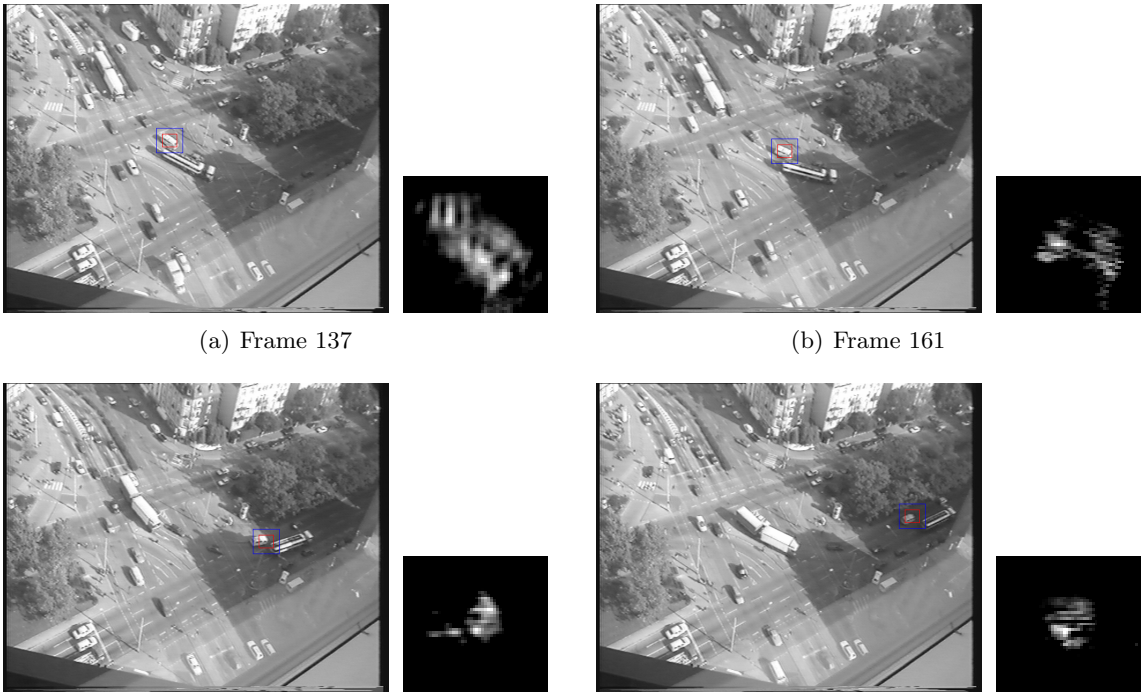
Figure 4: A successful tracking in case of a short occlusion.

# 6 Conclusions

This paper has explained ensemble tracking and Haar-like features and shown that this combination of two concepts improves the performance of a visual tracker in terms of illumination and occlusion problems which are very difficult to be solved. However, there are some limitations of this tracker so future works are necessary to fully solve a visual tracking problem.

## 6.1 Future Works

**Rotated Haar-like Feature** The set of Haar-like features used in this research is a subset of the extended set proposed by Lienhart and Maydt[11]. There are additional features, 45°rotated rectangles, in the set. The performance of object detection was improved by these rotated features. Although the result came from a different domain, it is strongly believed that the extended set is able to capture an object more accurately, thereby making a confidence map



(a) Frame 137 (b) Frame 161

(c) Frame 219 (d) Frame 270

Figure 5: A successful tracking in case of variant illumination.

more accurate in visual tracking.

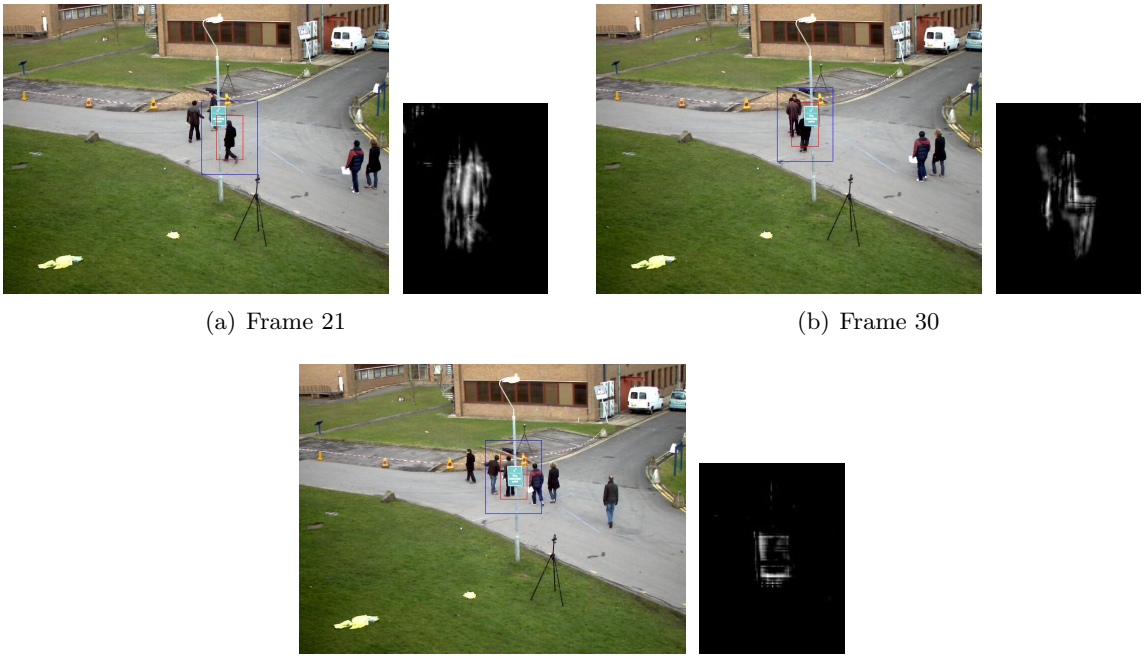
**Initialization of Tracking** This research assumes that the user initializes the location and the size of the target but the initialization is an essential component for fully autonomous visual tracking. A possible approach is the use of object recognition, which is also an elemental task of computer vision. If object recognition provides the location and the size of objects, this information can be used for the initial setting in visual tracking.

**Varying Size of Target** The target and background boxes are fixed once the user determined the initial size of the target box at the first frame in this research, because the size of the target in a frame is assumed to be invariant. However, there are many cases where the size changes in reality. For example, when objects become close to cameras, they appear larger and when they become far from cameras they appear smaller. For this reason, a method to estimate the varying size of target is necessary for successful visual tracking.

## References


[1] S. Avidan, "Ensemble Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261 –271, feb. 2007.

[2] O. Javed and M. Shah, "Tracking and Object Classification for Automated Surveillance," in *Proceedings of the 7th European Conference on Computer Vision*, 2002, pp. 343–357.



(a) Frame 21

(b) Frame 30

(c) Frame 65

Figure 6: Get stuck with an obstacle in case of a long occlusion


[3] O. Debeir, P. Van Ham, R. Kiss, and C. Decaestecker, "Tracking of Migrating Cells Under Phase-Contrast Video Microscopy With Combined Mean-Shift Processes," *IEEE Transactions on Medical Imaging*, vol. 24, no. 6, pp. 697 –711, june 2005.

[4] J. Yang, R. Stiefelhagen, U. Meier, and A. Waibel, "Visual Tracking for Multimodal Human Computer Interaction," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1998, pp. 140–147.

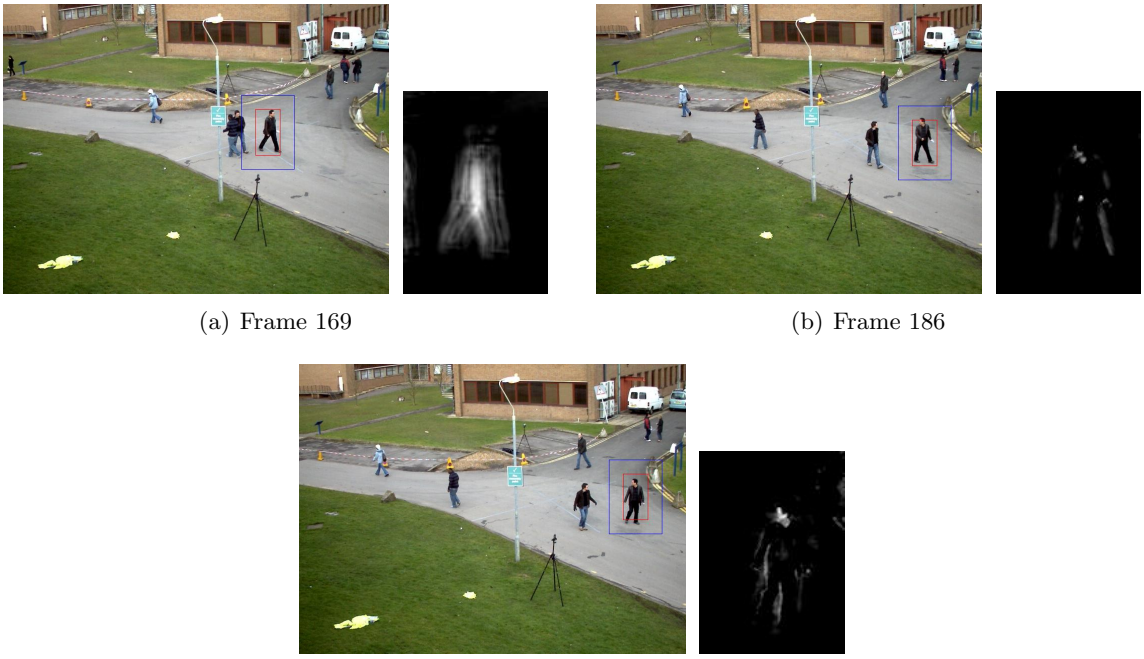
[5] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-Based Probabilistic Tracking," in *Proceedings of the 7th European Conference on Computer Vision*, 2002, pp. 661–675.

[6] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564 – 577, may. 2003.

[7] C. Yang, R. Duraiswami, and L. Davis, "Fast Multiple Object Tracking via a Hierarchical Particle Filter," in *the 10th International Conference on Computer Vision*, vol. 1, 2005, pp. 212–219.

[8] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 4, pp. 322 –336, apr. 2000.

[9] C. Papageorgiou, M. Oren, and T. Poggio, "A General Framework for Object Detection," in *the 6th International Conference on Computer Vision*, jan. 1998, pp. 555–562.



(a) Frame 169

(b) Frame 186

(c) Frame 188

Figure 7: Edge Phenomena: The confidence map becomes the edge of target as time goes by.


[10] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. I–511–I–518.

[11] R. Lienhart and J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection," in *Proceedings of International Conference on Image Processing*, vol. 1, 2002, pp. I–900–I–903.

[12] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," *the 9th European Conference on Computer Vision*, pp. 404–417, 2006.

[13] R. Collins, Y. Liu, and M. Leordeanu, "Online Selection of Discriminative Tracking Features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1631 –1643, oct. 2005.

[14] 11th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance(PETS 2009) Bechmark Data. [Online]. Available: http://www.cvg.rdg.ac.uk/PETS2009/a.html

[15] "Traffic Dataset from H.-H. Nagel in Universität Karlsruhe(TH)." [Online]. Available: http://i21www.ira.uka.de/image_sequences/