Object Tracking by Maximizing Classification Score of Detector Based on Rectangle Features

Akinori HIDAKA\textsuperscript{a}, Kenji NISHIDA\textsuperscript{††}, and Takio KURITA\textsuperscript{†,††}

SUMMARY In this paper, we proposed a novel classifier-based object tracker which combined a rectangular features based adaBoost detector with optical-flow based tracking method, Support Vector Tracker. We show that gradient of extended rectangular features can be calculated rapidly by using integral image method. The proposed tracker was tested on real video sequences. We applied our tracker for face tracking and car tracking experiments. Our tracker worked over 100fps while maintaining comparable accuracy to rectangle features based detector. The tracking routine without I/O process reaches 500 to 2500 fps with sufficient accuracy.

key words: Object Detection, Tracking, Support Vector Tracker, Rectangle Features, Boosting

1. Introduction

Efficient real-time tracking in complex environments is still a challenging task for the computer vision community. Target tracking is a key component in real-time applications such as surveillance, smart rooms, teleconferencing, and user interfaces. There are several approaches for tracking moving objects. Feature-based tracking relies on the persistence of image features such as image curves \cite{4}, \cite{9} or image appearances \cite{3}. Likewise, model-based tracking uses curves \cite{11}, \cite{18} or appearance \cite{3}. Particle filters \cite{8}, \cite{12} or mean shift methods \cite{2}, \cite{5} are often used to treat target dynamics. For face tracking in a crowded scene, however, the tracking system needs to rely more on target representation than on target dynamics.

The computational complexity of the target tracker is critical for most applications. Only a small percentage of system resources is allocated for tracking because the rest must be assigned to high-level tasks such as recognition, trajectory interpretation, and reasoning. Thus, it is important to develop a target tracking algorithm based on fast and accurate object detectors. Recently, Avidan \cite{1} proposed a tracking algorithm called Support Vector Tracker (SVT) based on Support Vector Machine (SVM) \cite{13}, \cite{14}. SVT is to integrate a classification approach which uses class-specific information and the framework of optical-flow based tracking that is comparatively unstable. However, kernel SVM constructed by raw intensity vectors is very time-consuming and sensitive against noises or change of target’s view.

On the other hand, currently a lot of useful image features are studied. Rectangular features (RF) by Viola-Jones \cite{15}, \cite{16} is one of such a features. They introduced a new image presentation called Integral Image which can calculate feature value of RF very quickly.

The combination of rectangular features and optical-flow framework might make Avidan’s SVT more accurate and fast. Although SVM using raw intensity vectors is weak for illumination change or partial occlusion, RF based classifier is comparatively robust for that effect because RF uses contrast information in local region, and the output of classifier is determined by majority vote of decisions in such a local region.

In this paper, rectangular features are extended as differentiable to extract optical-flow-like information. And we show that such information can be calculated rapidly by only using integral image method.

As a result, proposal tracker work very fast and accurate; entire speed is 115fps and tracking routine without I/O process runs 590fps while maintain same level accuracy to Viola-Jones detector.

2. Boosted Detector Constructed by Rectangle Features

The face detector proposed by Viola and Jones \cite{15}, \cite{16} became very successful because of its robustness and high speed. We used three key features of their work, integral image, rectangle features, and boosting, in our face detector. We replaced the binary classification function by the sigmoid function because the derivative of the classification function is needed for our tracking algorithm. Details of these features are described in this section.

2.1 Integral Image

The integral image method, also called summed area table \cite{6}, is used to reduce the computational cost of calculating the mean intensity in a rectangular region. The integral image is a reference table which consists
A rectangle feature consists of two to four small rectangles. RFs has four degree of freedom depending on the location and size of region paying attention to.

The mean intensity of rectangular regions of any size and position can be calculated from only four table references.

2.2 Rectangular Features

A rectangle feature consists of $n$ small rectangles that are neighboring each other and are same size (Fig. 1). The small rectangles $\{s_k\}_{k=1}^{n}$ are defined as

$$s_k = \{(x,y)|x_k^- \leq x \leq x_k^+, y_k^- \leq y \leq y_k^+\}.$$

It is supposed that $\{s_k\}_{k=1}^{n}$ sufficient $\bigcup s_k = q$, and $s_i \cap s_j = \phi$ for $i \neq j$. The mean intensity of an image patch $I_{s_k} = \{i(x,y)|(x,y) \in s_k\}$ is defined as

$$m(I_{s_k}) = \frac{1}{wh} \int_{s_k} i(x,y) dx \, dy = \frac{1}{wh} \int_{y_k^+}^{y_k^-} \int_{x_k^-}^{x_k^+} i(x,y) dx \, dy,$$

where $w$ and $h$ are the width and the height of the small rectangles, respectively.

Each $s_k$ has a subtraction flag $c_k \in \{-1,+1\}$ (small rectangles which are assigned $c_k = +1$ are illustrated as gray/black rectangles in Fig. 1, respectively). We represent a single rectangle feature as $r = \{s_k, c_k\}_{k=1}^{n}$. The feature value is obtained from the difference of the mean intensities between the gray and black small rectangles:

$$f(r, I_q) = \sum_{k=1}^{n} c_k m(I_{s_k}).$$

Originally, Viola and Jones used a binary threshold function for the classification function:

$$h(r, I_q) = \begin{cases} 1 & (I_q \text{ is face}) \text{ if } pf(r, I_q) > p\theta \\ 0 & (I_q \text{ is nonface}) \text{ otherwise } \end{cases},$$

where $p$ and $\theta$ are determined from machine learning. We replaced the classification function with the sigmoid function to enable the differentiation.

$$h(r, I_q) = \frac{1}{1 + e^{-p(f(r, I_q)-\theta)}}$$

2.3 Boosting

Boosting is a learning algorithm that integrates multiple classifiers to yield a stronger classifier. The obtained classifier is called a strong classifier, and the classifiers used for the components of the strong classifier are called weak classifiers. In Viola and Jones’ method, a weak classifier is a single rectangle feature.

The training algorithm of adaboost is as follows.

- Enter training set $\{(x_i, t_i)\}_{i=1}^{N}$ ($t_i = 1 \iff x_i$ is face, $t_i = 0 \iff x_i$ is nonface).
- If $t_i = 1$ then $w_{1,i} = \frac{1}{N}$ else $w_{1,i} = \frac{1}{N}$ ($p$ and $q$ is a number of faces and nonfaces, respectively).
- For $t = 1, \cdots, T$
  - $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{i=1}^{N} w_{t,i}}$
  - For each rectangle feature $r^{(j)}$, optimize parameters $p$ and $\theta$ under the weighted error
    - $e^{(j)} = \sum_{i}^{N} w_i | h(r^{(j)}; x_i) - y_i |$
    - $j^* \leftarrow \arg\min \{e^{(j)}\}$, $\epsilon_{\min} = e^{(j^*)}$; $r_t \leftarrow r^{(j^*)}$.
    - $\alpha_t = \log\left(\frac{1-\epsilon_{\min}}{\epsilon_{\min}}\right)$.
    - $w_{t+1,i} = w_{t,i} (\frac{1-\epsilon_{\min}}{\epsilon_{\min}})^{1-h(r_t,x_i)-t_i}$.

The final classification function is

$$H(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^{T} \alpha_t h(r_t, x) \geq \Theta \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

where $\Theta$ is a threshold determined manually.

3. Object Tracking by Maximizing Score Function

In the framework of detection-based object tracking [1], a tracker follows targets by maximizing the score function obtained from the target detector. It can be expected that the target’s location has the highest score in the image. If the target moves anywhere, the peak of the score will synchronously move. Therefore, the target tracking problem can be treated as numerical optimization problem for the target’s score function. The derivative of the score function is helpful to solve such a problem: When the target slightly moves from $q$, the target is located in a direction of the gradient of the score function at $q$.  

Fig. 1 Rectangle features (RFs). One RF is constructed from two to four small rectangles. RFs has four degree of freedom depending on the location and size of region paying attention to. For example, there are about 130,000 different RFs in 24 $\times$ 24 pixels region.
3.1 Score Function and Target Function

In this paper, the score function $E$ is obtained from rectangle features based detector. $E(I)$ is defined from Eqs. 3 and 4, as

$$E(I) = \sum_{t=1}^{T} \alpha_t h(r_t, I) = \sum_{t=1}^{T} \frac{\alpha_t}{1 + e^{-p_t(f(r_t, I) - \theta_t)}}. \quad (5)$$

Let us consider images $I'$ and $I$, which are consecutive frames in a video sequence including target and other objects. Suppose that the target was located at $q$ in the past frame $I'$, and moved to unknown region $q_f$ in the current frame $I$. Let $q_f$ be an image patch of the current frame $I$ at $q$.

Now, we consider how $E(q_f)$ changes in a neighboring region of $q$. Let $g(u, v)$ be a region that is $(u, v)$ distant from $q$. To reduce computation, $I_{q(u, v)}$ is approximated by a first-order Taylor expansion at $q$ as follows:

$$I_{q(u, v)} \simeq I_q + uI_{qx} + vI_{qy},$$

where $I_{qx} = \frac{\partial I_q}{\partial x}$, and $I_{qy} = \frac{\partial I_q}{\partial y}$. From this, Eqs. 1 and 2 are approximated as follows:

$$m(I_{sk}) \simeq m(I_q) + um_x(I_{sk}) + vm_y(I_{sk})$$

$$f(r_t, I_q) \simeq \sum_{k=1}^{n} c_k \{m(I_{sk}) + um_x(I_{sk}) + vm_y(I_{sk})\}$$

$$f^*(r_t; u, v)$$

where $m_x = \frac{\partial m}{\partial x}$, and $m_y = \frac{\partial m}{\partial y}$. Replacing $f$ with $f^*$ in Eq. 5, we obtain the approximation of $E(I_q)$ which is the target function to be maximized. It is defined using the movement $(u, v)$ as follows:

$$E^*(u, v) = \sum_{t=1}^{T} \frac{\alpha_t}{1 + e^{-p_t(f^*(r_t; u, v) - \theta_t)}}. \quad (6)$$

If the assumption that $E^*$ as well as $E$ becomes maximum at $q_f$ is correct, the target will be in $q_{(u_m, v_m)}$ where $(u_m, v_m)$ is a maximizer of $E^*(u, v)$. Therefore, our tracking task is now forming a maximization problem of $E^*(u, v)$. This problem is solved by some numerical analysis algorithm. For example, the steepest descent method is applied as follows:

$$\frac{\partial E^*}{\partial u} = \sum_{t=1}^{T} \frac{\alpha_t e^{-p_t(f^*(r_t; u, v) - \theta_t)}(1 + e^{-p_t(f^*(r_t; u, v) - \theta_t)})^2 p_t f_{tu}^*}{\partial u}$$

$$\frac{\partial E^*}{\partial v} = \sum_{t=1}^{T} \frac{\alpha_t e^{-p_t(f^*(r_t; u, v) - \theta_t)}(1 + e^{-p_t(f^*(r_t; u, v) - \theta_t)})^2 p_t f_{tv}^*}{\partial v}(u, v) \leftarrow (u + \eta \frac{\partial E^*}{\partial u}, v + \eta \frac{\partial E^*}{\partial v}),$$

where

$$f_{tu}^* = \frac{\partial f^*(r_t; u, v)}{\partial u} = \sum_{k=1}^{n} c_k m_x(I_{sk}),$$

$$f_{tv}^* = \frac{\partial f^*(r_t; u, v)}{\partial v} = \sum_{k=1}^{n} c_k m_y(I_{sk}),$$

and $\eta$ is a step length that is selected manually.

3.2 Multilevel Blurred Images to Avoid Local Maxima

Multimodality of the target function might cause to converge on local maxima. To avoid this problem, we used multilevel blurred images, like Avidan’s pyramid images [1]. In a blurred image, although the precise position of the true maximum becomes slightly unclear, the multimodality of the target function is toned down. Therefore, by using the target’s location in the previous frame as the initial position, we start the maximization procedure at the most strongly blurred image. And, the position of the maximum is used as the start position for the next level. In this way, our tracking algorithm gradually approaches the true maximum while avoiding local maxima. The maximum’s position in the original (unblurred) image is considered to be the new position of the target.

The tracking algorithm is as follows. (It is assumed that a number $m$ and initial guess $q^m$ were given appropriately.)

<table>
<thead>
<tr>
<th>Table 1 Tracking Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter $I^m$ (m-th image) and $q^{m-1}$ (target position in previous frame).</td>
</tr>
<tr>
<td>2. Create an L-level blurred images ${I^m_l}_{l=1}^{L}$ from $I^m$ ($I^m_1$: original image, $I^m_L$: most strongly blurred image).</td>
</tr>
<tr>
<td>3. Set $q^m_0 \leftarrow q^{m-1}$.</td>
</tr>
<tr>
<td>4. For $l = L, \cdots, 1$</td>
</tr>
<tr>
<td>- Maximize Eq. 6 in blurred image $I^m_l$ starting position $q^m_0$; let $q^m_{l-1}$ be the position of the maximum.</td>
</tr>
<tr>
<td>5. Output $q^m_0$ as the position of the tracking result.</td>
</tr>
<tr>
<td>6. Set $m \leftarrow m + 1$; Go to 1.</td>
</tr>
</tbody>
</table>
4. Experiments

In this section, we describe face tracking and car tracking experiments.

4.1 Face Tracking

Face detector used here was trained by boosting of 200 iterations. 300 frontal faces and 700 background images of $24 \times 24$ pixel were used for training. These training samples were collected from internet. The target video is filmed by us for the experiment. The video includes 1,060 frames which are $320 \times 240$ pixel. Each frame includes one frontal or half-profile face to be tracked. The target moved around in a room that contained bookshelves and a door in the background while maintaining nearly same size.

We compared the performance of the proposed tracker, called Steepest Descent Method (SDM) tracker, with a baseline tracker. The baseline tracker is called Neighborhood Search (NS) tracker which performs exhaustive search in local $N \times N$ area around previous target position ($N$ is predefined number). Each algorithm has following parameters:

- **common**: blur level $L$
- **SDM**: iteration number $\tau$, step length $\eta_l$ at $l$-level blurred image,
- **NS**: search area range $N_l \times N_l$ at $l$-level blurred image.

We optimized these parameters in manually. For SDM and NS tracker, we test many combination of parameters, and found the best ones. The best and some examples of tested parameters are shown in Fig. 4.

SDM tracker work at 113 frames per second (fps) with the best parameters, near four times faster than video rate (30fps). However, actually 80% of processing time, 7.6sec of 9.4sec for 1060 frames, was spent for the routine unrelated to tracking, mainly image input/output routine. Excepting the routine not related to tracking, our tracking routine (4. of Fig. 1) ran
at 591fps against 263fps of the tracking routine of NS tracker.

Tracking accuracies of both trackers are shown in Figs. 2 and 3. These figures are comparison of tracked position with ground truth. The gradient of curves in Fig. 2 implies the moving speed of target. Our SDM tracker worked as well as NS tracker, so SDM tracker maintain same level accuracy to Viola-Jones detector.

4.2 Car Tracking

Car detector used here is trained by adatboost of 200 steps. For the training, 144 front part of truck images and 1,000 background images were used. The size of each images is $32 \times 32$ pixel. Truck images for training were collected by cropping from the video sequence constructed from 8,464 frames real world scene. This video was filmed at Bayshore Route, a highway of Tokyo. The test sequence is first 650 frames in the video. This sequence always includes one front part of a truck which keeps almost same size. The first half scene is loose curve and then target moves slowly. The last 100 frames are the scene of lane change, so the target moves slightly fast.

We test the video by SDM tracker. The results are showed in Figs. 5 and 7. For the case that movement of the target was small, our tracker was allowed very economical parameters: $L = 1$, $\tau = 5$. In spite of considerably low computational cost, SDM tracker well converged on the target and failure did not occur. Processing speed of the tracking routine without I/O process in this case is extremely fast, over 2,500fps.

4.3 Tracking Failure

The test sequence for face tracking experiment includes two difficult scene. First scene is about from 700th to 800th frame. In this scene, target panned over left and right and then became half-profile (top-right and bottom-left image in Fig. 6). Second is the last 100 frames, target moved comparatively fast and slightly rotated (bottom-center and bottom-right image in Fig. 6). In these scenes, a reliability of face detector becomes less, and then false positives will easily occur. The trackers No. 1, 3, 5, 7 in Fig. 4, which mistracked the target, converged on such a false maxima.

5. Limitation and Future Work

We proposed a novel classifier-based tracker which combined a rectangle features based adaboost detector with SVT. Proposal RF-based SDM tracker was tested on real video sequences. Our SDM tracker worked over 100fps while maintaining comparable accuracy to RF-based detector. The tracking routine without I/O process runs 500 to 2500 fps with sufficient accuracy.

However, our tracker has some limitations and inadequacies. First, the speed of SDM tracker weakens gradually when the number of targets in an image increases. Second, SDM tracker can not handle appearance/disappearance of target. Third, there is no clue to identify tracked target from multiple targets in current our tracker.

In the future work, we will study about integration of this method and temporal filtering. It is a combination of three existing tracking approaches: classifier, optical-flow and temporal filtering based approach.

References

The results of the car tracking. The movie consists of 650 frames. (Left) $x$ coordinate of tracked positions. Solid line shows results and dotted line shows ground truth. (Right) Deviation from the ground truth. In this experiment, target mostly moved small. Therefore, SDM tracker required only unblurred images and 5 times iteration to obtain the failure-free result. Then the tracking routine in SDM tracker run over 2,500fps.


Akinori Hidaka received the B.Eng degree from Ibaraki University and the M.Eng. degree from the University of Tsukuba, in 2004 and 2006, respectively. He is currently a student of Dr. Kurita in University of Tsukuba. His current research interests include object recognition and tracking based on statistical pattern recognition. He is a member of IEICE.

Kenji Nishida received the B.Eng. degree from Keio University in 1981. He joined the Electrotechnical Laboratory, Japan in 1981. From 1985 to 1989 he was a visiting research scientist at Institute for new generation COmputer Technology. He is currently chief researcher of Neuroscience Research Institute, National Institute of Advanced Industrial Science and Technology (AIST). His current research interests include brain-like information processing and statistical pattern recognition. He is a member of IPSJ and JNNS.

Takio Kurita received the B.Eng. degree from Nagoya Institute of Technology and the Dr.Eng. of University of Tsukuba, in 1981 and 1993, respectively. He joined the Electrotechnical Laboratory, AIST, MITI, Japan in 1981. From 1990 to 1991 he was a visiting research scientist at Institute for Information Technology, NRC, Ottawa, Canada. He is currently Deputy Director of Neuroscience Research Institute, National Institute of Advanced Industrial Science and Technology (AIST). His current research interests include statistical pattern recognition and neural networks. He is a member of IEEE Computer Society, IEICE, IPSJ, JNNS, the Behaviormetric Society of Japan, and Japanese Academy of Facial Studies.
Fig. 6  Examples of the face tracking result by SDM tracker. White boxes in images show tracked position.

Fig. 7  Examples of the car tracking result by SDM tracker. White boxes in images show tracked position.