Adaptive Compressive Tracking Based on Perceptual Hash Algorithm

Lei ZHANG, Zheng-guang XIE* and Hong-jun LI
School of Electronic Information, Nantong University, Nantong 226019, China
*Corresponding author

Keywords: Compressive Tracking, Perceptual Hash Algorithm, Real-time Performance.

Abstract. As the compressive tracking algorithm is easily failed to track target due to the classifier learning rate is fixed when appearances and lightings of target get seriously changed and completely loses the target after heavy occlusion, we propose adaptive compressive tracking algorithm based on perceptual hash algorithm. It makes the compressive tracking algorithm more adaptive to the change of target appearances by adjusting the learning rate of classifier in real time according to the Hamming distance between the hash fingerprints of current target and the original one. Experimental results show that the method can quickly and accurately track the target in the case of the object appearances change and the object is occluded. Our tracker accuracy and robustness have been improved and real-time performance.

Introduction

In recent years, with the development of computer video processing technology, computer vision has become a hot research topic [1]. As an extremely important part of computer vision, moving object detection and tracking technology has been widely studied and applied to video surveillance, image processing, artificial intelligence and so on[2-3]. Compared to the early target tracking algorithms, such as mean filtering, particle filtering, optical flow and template matching method, scholars have put forward a lot of effective target tracking algorithms that have made a great breakthrough. However, the object tracking is still challenging because of the change of target pose, motion blur, the change of illumination and especially heavy occlusion.

At present, the research focus of real-time target tracking is the discriminative algorithm [4] that poses the tracking problem as a binary classification task. By choosing the appropriate classifier, it classifies the candidate image blocks and select the image block with the maximum confidence as the target location. The target model classifier updates later to improve the tracking performance. Grabner et al. [5] proposed an online boosting algorithm to select features for tracking. However, it only uses one positive sample (i.e, the current tracker location) a few negative samples while updating the classifier. As the appearance model is updated with noisy and potentially misaligned examples, this often leads to the tracking drift problem. Babenko et al. [6] introduced multiple instance learning into online tracking where samples are considered within positive and negative bags or sets in order to solve the problem of classifying samples wrong. However, due to the complexity of calculation, it can’t meet the requirements of real-time. In [7] an algorithm named online discrimination feature selection is proposed. Hare et al. [8] demonstrated a method that classifies target and background by using support vector machine algorithm works well in tracking. Zhang et al. [9-10] proposed an efficient target tracking algorithm based on compressive sensing theory. It takes the generalized haar-like features of target and background as the positive and negative samples of online learning to update classifier, which greatly reduces the complexity of tracking algorithm and has good real-time performance. However, while occlusion occurs in tracking, all of the above methods are easy to lose the true target by regarding the background as object, which leads to failure of tracking. It is required to design an object tracking method to deal with this problem.
**Proposed Method**

The deficiency of the discriminative algorithm, such as compressive tracking algorithm, is that the update rate of classifier fixed. It doesn’t know whether or not the current tracking target is true. The target template will be incorrectly updated with background when the target is obscured, which leads to drift or loss of target and tracking failure after long occlusion. In this paper, we combine the compressive tracking with perceptual hash algorithm to improve its performance for occlusion.

The pHash algorithm is mainly used in similar image search. It generates a fingerprint string for each image to determine whether the images are similar by comparing Hamming distance between fingerprints. The pHash Algorithm procedure is as follows:

1) Resize the dimension of target image to 32*32;
2) Use discrete cosine transformation (DCT) on the gray image to get the DCT coefficient matrix with the size 32*32;
3) Preserve the 8*8 matrix of the upper left corner and calculate the mean value of DCT matrix;
4) Set values of the 8*8 matrix which are greater than or equal to the mean value to 1 and others to 0. The compositor of the matrix makes up a 64 bit integer, which is the fingerprint of the image.

As long as the overall structure of the image remains unchanged, the fingerprint is unchanged. The different bits, i.e. the Hamming distance $d$ between two strings shows the similarity degree of two images. $d = 0$ means two images are very similar while $d > 10$ means the difference of two images are more than fifty percent, which is verified massive experiments. The tracking algorithm is able to determine whether the current target is the target of interest and adjust the update rate $\lambda$ of classifier real-time in Eq. 1 to keep the target template correct.

$$
\lambda = \begin{cases} 
\lambda_1, & d < 5 \\
\lambda_2, & 5 \leq d < 10 \\
1, & d \geq 10 
\end{cases} 
$$  (1)

The classifier stops updating while heavy occlusion occurring ($d > 10$), which makes it possible to locate the real object after occlusion, and updates quickly when target changes slowly ($d < 5$). This method effectively improves the accuracy of target tracking with occlusion. Our algorithm are summarized as follows:

**Input:** t-th video frame
1) Calculate and save the hash fingerprint $H_{t-1}$ of the target at the (t-1)-th frame.
2) Sample a set of image patches, $D' = \{z | \|l(z) - l_{t-1}\| < \gamma\}$ where $l_{t-1}$ is the tracking location at the (t-1)-th frame, and extract the features with low dimensionality.
3) Use trained classifier to each feature vector $v(z)$ and find the tracking location $l_t$ with the maximal classifier response.
4) Calculate and save the hash fingerprint $H_t$ of the current target.
5) Adjust Hamming distance $d$ between $H_{t-1}$ and $H_t$, and $\lambda$ as (1).
6) Sample two sets of image patches $D' = \{z | \|l(z) - l_t\| < \alpha\}$ and $D'' = \{z | \|l(z) - l_t\| < \beta\}$ with $\alpha < \zeta < \beta$.
7) Extract the features with these two sets of samples and update the classifier.

**Output:** Tracking location $l_t$, learning rate $\lambda$ and classifier parameters.

**Experimental Results**

We perform extensive evaluations on VOT 2014 dataset to validate our approach. We compared with the compressive tracking (CT) [9] and the fast compressive tracking (FCT) [10] to verify the
effectiveness and robustness of the proposed algorithm in tracking with heavy occlusion. The search radius for drawing positive samples is set to $\alpha=4$ which generates 45 positive samples. The inner and outer radii for the set $X_\zeta^\beta$ that generates negative samples are set to $\zeta=8$ and $\beta=30$, respectively. We randomly select 50 negative samples from set $X_\zeta^\beta$. The search radius for set $D_\gamma$ to detect the object location is set to $\gamma=20$ and about 1100 samples are generated. And the learning rate $\lambda_1$ is set to 0.35 and $\lambda_2$ is set to 0.85. The computer is configured to Intel Core i3-2120 processor, 2GB ram, Win7 32 operating system, and the software environment is MATLAB 2011b.

In order to objectively compare the tracking performance of three tracking algorithms, the success rate (SR) and center location error (CLE) are used to evaluate the tracking results. Center error represents the mean value of the Euclidean distance between the position center of tracking and the center of the actual target location. The success rate is defined as: $score = \frac{\text{area}(\text{ROI}_T \cap \text{ROI}_G)}{\text{area}(\text{ROI}_T \cup \text{ROI}_G)}$ where $\text{ROI}_T$ means the tracked region of current frame and $\text{ROI}_G$ means the actual region. The tracking of current frame is successful while $score \geq 0.5$. The average results of 50 times simulation are shown in Table 1. Bold fonts indicate the best performance. It’s easy to find that the performance of our algorithm has improved significantly in tracking possible to locate the true object after heavy occlusion according to the result that shown in Figure 1.

**Conclusion**

In this paper, a simple and efficient method based on perceptual hash algorithm is proposed for object tracking when occlusion occurs frequently. The performance of the compressive tracking algorithm in tracking that the object is obscured or lost is evidently improved by adjusting the learning rate of classifier in real time, which makes the target template more adaptive and proper. We have successfully achieved the results that the method works quickly and accurately when target is occluded or lost and is able to locate the real object after heavy occlusion. Robustness is significantly improved.

**Acknowledgments**

This work was supported by the National Natural Science Foundation of China (NO. 61171077, the University Science Research Project of Jiangsu Province (NO. 16KJB510036), the Science and Technology Program of Nantong (NO. MS12016025), Nantong University Undergraduate Training Program for Innovation (NO. 2016069).

<table>
<thead>
<tr>
<th>Video sequences</th>
<th>CLE</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ours</td>
<td>CT</td>
</tr>
<tr>
<td>Coke</td>
<td>40</td>
<td>51</td>
</tr>
<tr>
<td>Coupon</td>
<td>13</td>
<td>19</td>
</tr>
<tr>
<td>David</td>
<td>16</td>
<td>22</td>
</tr>
<tr>
<td>Faceocc1</td>
<td>20</td>
<td>31</td>
</tr>
<tr>
<td>Faceocc2</td>
<td>14</td>
<td>22</td>
</tr>
<tr>
<td>Subway</td>
<td>8</td>
<td>67</td>
</tr>
<tr>
<td>Suv</td>
<td>23</td>
<td>36</td>
</tr>
<tr>
<td>Sylvester</td>
<td>14</td>
<td>21</td>
</tr>
<tr>
<td>Tiger1</td>
<td>30</td>
<td>38</td>
</tr>
<tr>
<td>Walking2</td>
<td>29</td>
<td>51</td>
</tr>
</tbody>
</table>
Figure 1. Screenshots of some sampled tracking results.

References


