Adaptive Learning Image Tracking Algorithm Based on Characteristic Fusion

Yu Feng¹, Chenlong Deng¹, Liyang Wu¹, Xingdi Wang¹, Jing Xu† and Lei Liu†
¹ College of Artificial Intelligence, Nankai University, No. 38 Tongyan Road, Haihe Education Park, Tianjin 300350, CN

1813632@mail.nankai.edu.cn; deng_chenlong@mail.nankai.edu.cn; 1813575@mail.nankai.edu.cn; 1811620@mail.nankai.edu.cn; xujing@nankai.edu.cn; liulei@nankai.edu.cn

Abstract. To solve the problem of model free real-time image tracking, this paper proposes an image tracking algorithm based on characteristic fusion to adaptively learn the historical characteristics of the target (CFAL). In this algorithm, the color feature is redesigned by using the idea of hog to achieve the feature consistency; the particle filter is combined with the reality constraint to improve the matching efficiency. Finally, the algorithm is programmed and tested in reality, animation and video scenes. The experimental results show that adaptive feature learning and weight adjustment can improve the tracking effect and CFAL can track the target stably and reliably under the environment of attitude change and partial occlusion.

1. Introduction
Image tracking has always been a problem in the field of computer vision [1, 2]. Image tracking is to find the target in the subsequent frames after knowing the target position in the first frame. The lack of prior knowledge, the occlusion of the target, the disappearance of the target, the motion blur, the drastic changes of the target and the environment, the high-speed movement of the target, and the shaking of the camera will all have an impact on target tracking. These factors have made image tracking a difficult problem in the CV field. In this paper, an adaptive learning target tracking algorithm based on feature fusion is proposed, which we call CFAL. The algorithm mainly includes feature extraction, target matching and adaptive learning. For a set of image sequences, multi-dimensional image features are extracted and fused [3, 4]. Based on feature matching degree [5] and practical constraints, the region with the highest matching degree is taken as the target. Thus, the tracking target is identified from the image of the next frame. The identified target image is used as new information for adaptive learning to improve the generalization of target features.

2. Extraction and fusion of multi-dimensional features

2.1. Histogram of color features: Lab spatial color distribution
Lab [6] color space has perceptual uniformity and Euclidean distance invariance. Therefore, using the color channel of Lab to extract the color histogram of the image, as the color feature of the image, it not

* These authors are joint corresponding authors (Prof. Jing Xu, Dr. Lei Liu).
only conforms to the true perception of colors, but also reduces the color from the three dimensions of RGB to the two dimensions of a and b.

2.1.1. Production of histogram of color distribution. For each block in the image, traverse the pixels in the block, and each pixel can be assigned to a bin[7]. Finally, the number of pixels in each bin is counted and normalized to generate a color feature histogram. The histogram reflects the color distribution of the block.

2.1.2. Process of color feature extraction. We divide the matching window into cells. Four cells constitute a block. From left to right, from top to bottom, the color histogram of the block is extracted and stitched by cells. Finally, we can get a feature vector group describing the color of the matching region.

2.2. Histogram of edge features: directional gradients
Edge feature extraction, uses the classical direction gradient histogram (HOG)[8, 9, 10] method. The specific content will not be expanded here. The key is to unify the form of edge features and color features to facilitate feature fusion.

2.3. Feature matching method
2.3.1. Block matching. Each block obtains a set of normalized feature histograms as the distribution of features. The concept of KL divergence[11] is introduced to evaluate the similarity of feature distributions between two blocks.
KL divergence is a tool used to describe the difference between two distributions. For $D_{KL}(p \parallel q)$, the higher the similarity, the smaller the value. In order to make the similarity positively correlated with the final output, a Gaussian mapping function is added.
The matching degree function between two blocks is defined as:

$$r^{k_v}(X_t^k, X_{t+1}^k) = e^{-\frac{D_{KL}(X_t^k, X_{t+1}^k)^2}{2\sigma^2}} \times 100\%$$

where $k$ represents the k-th block, $v$ represents the feature type, and $t$ represents the sampling time point.

2.3.2. Feature matching degree. For a set of matching windows $(X_i, X_{i+1})$, With $m \times n$ blocks. The matching degree of $k_{ij}$ blocks can be described as $r^{v}(k_{ij}, i=1,2,...,m, j=1,2,...,n, v$ represents the feature type. Considering that edge pixels are more susceptible to occlusion or background pixels interference, the image features at the center position will be more stable. Therefore, we define the weight of blocks as $\gamma(i, j)$, among which, the block matching degree located in the center of the image has higher weight and the edge is lower. Define a single feature matching degree as follows:

$$R^{v}(X_t^i, X_{t+1}^i) = \sum_{i=1}^{m} \sum_{j=1}^{n} \gamma(i, j) r^{v}(k_{ij}, X_t^k, X_{t+1}^k)$$

2.4. Adaptive learning feature fusion
The matching degree of color feature and edge feature can be obtained for a set of matching window $(X_i, X_{i+1})$. Two features have different distinguishing abilities for different scenes. Therefore, it is necessary to dynamically adjust the feature weight of the next frame and realize adaptive feature fusion based on the matching degree of the current frame[12].
The adaptive fusion weight function is defined as follows:
Among them, the $\tau$ is an adaptive learning rate used to control the speed of adjusting the weights. Then the region matching degree can be expressed as:

$$R(X_t, X_{t+1}) = \beta_t R\left(X^v_t, X^v_{t+1}\right) + (1 - \beta_t) R\left(X^w_t, X^w_{t+1}\right)$$

where the $v$ represents the color feature and the edge feature.

2.5. Feature matching process[13]

1. At $t$ moment, the target in image $S_t$ coordinates is $(x_t, y_t)$.
2. Generate $k$ particle coordinates: $(\hat{x}^i_{t+1}, \hat{y}^i_{t+1}), i = 1, 2, \cdots, k$.
3. Generate corresponding matching window $X'_{t+1}$ for every particle.
4. Extract color and edge features from matching window $X'_{t+1}$ to match sample features and obtain matching $R'_{t+1}$.
5. Select the particle with the highest matching degree as the target position for the next frame:

$$\left(\hat{x}_{t+1}, \hat{y}_{t+1}\right) = \left(\hat{x}^j_{t+1}, \hat{y}^j_{t+1}\right)$$

The process is shown in the figure 1.

3. Adaptive learning

$$\beta_{t+1} = \beta_t + \tau \left( e^{R(X_{t-1}^v, X_t^v) - \sum_{n=1}^{N} \frac{R(x_{t-1}^v, \hat{x}_{t}^n)}{e^{\sum_{n=1}^{N} R(x_{t-1}^v, \hat{x}_{t}^n)}} - \beta_t \right)$$ (3)

Without providing sufficient data sets in advance, we can only improve the amount of information by learning. When the target window of the $t$ time is $X_t$, characterized as $W_t$, the change of the sample feature in the following process is shown in the figure 2.

Since the color and edge features are linearly superimposed as we considered, the features are updated by:

$$W_{t+1}^T = (1 - \alpha)W_t^T + \alpha W_t$$ (5)

Among them, $\alpha$ is the learning rate of feature and describes the speed of feature learning.

4. Process of adaptive learning image tracking algorithm based on characteristic fusion (CFAL)

1. Obtain initial feature $W_0^T$, initial target position $(x_0, y_0)$.
2. Initialize feature fusion weight $\beta_0 = 0.5$.
3. Define threshold diameter $D$, adaptive learning rate $\tau$, characteristic learning rate $\alpha$.
4. loop:
a. Match feature following Section 2.5.
b. Update feature fusion weight satisfying equation (3).
c. Learn characteristics according equation (5).

In order to achieve better results, we used several tricks inspired by intuition.

4.1. Weight the particle generation range based on the target moving speed.

To predict the target position, the speed of the target estimated through historical information can be utilized. Based on the magnitude and direction of the velocity, the particle generation range is weighted. The specific weighting method is shown in figure 3.

4.2. Suppress the influence of background on color characteristics.

Considering that there is a part of the background in the window, in order to suppress the influence of it on color feature extraction, the surrounding area $S$ outside the matching window is examined, and the color feature histogram of $S$ is extracted. The schematic diagram is shown in figure 4.

5. Experimental Results and Analysis

5.1. Evaluation criteria

As figure 5, mark the correctly recognized target area as TP. Unrecognized target area is marked as FN. The area mistaken for the target area is marked as FP. We can evaluate the tracking effect by Precision Rate and Recall Rate. To evaluate comprehensive performance, we use the harmonic mean of Precision Rate and Recall Rate.

5.2. Tracking effects for various situations

5.2.1. Morphological changes. In order to test the robustness of CFAL to the morphological change of the target, we selected a real scene (case 1) to track the shoes, an animation scene (case 2) to track an rotating target. The tracking results are shown in the figure 6, 7, 8, 9 (the image sequence is obtained by sampling frames, not a complete sequence, only the tracking process is shown). From the actual results, it is possible to track the target whose shape has changed greatly. The more drastic the change of the attitude, the lower the tracking accuracy, but a stable tracking effect can still be maintained.
5.2.2. Consider occlusions. In order to test the tracking effect of CFAL when the target is partially occluded, two groups of image sequences are selected for tracking: one group reflects the process of the mobile robot passing through the obstacle (case 3), the other group is the segment selected from the documentary (case 4), tracking one of them from a group of wild animals. The experimental results are shown in the figure 10, 11, 12, 13 (the image sequence is obtained by frame extraction, not a complete sequence, only showing the tracking process). From the tracking effect, the algorithm has a certain resistance to partial occlusion. If the parameters are adjusted properly, the moving trend of the target can be tracked, but the size of the matching window will fluctuate.
5.3. Effect of Each Parameter on The Tracking Effect

In order to study the influence of single parameters on tracking effect, the selected image sequence case1 is tracked by different size parameters, and the harmonic average of precision and recall of the last 50 frames is calculated.

Table 1.

| τ  | 0   | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1   |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Precision avg | 69.1% | 77.5% | 74.5% | 87.7% | 89.3% | 93.5% | 68.5% | 70.0% | 63.3% | 79.4% | 75.5% |
| Recall avg    | 51.8% | 47.4% | 85.4% | 69.5% | 70.5% | 67.8% | 48.0% | 72.2% | 64.7% | 67.7% | 74.5% |

Table 2.

| α  | 0    | 0.05 | 0.1  | 0.15 | 0.2  | 0.3  | 0.4  | 0.5  | 0.6  | 0.8  | 1    |
|----|------|------|------|------|------|------|------|------|------|------|------|
| Precision avg | 92.9% | 91.5% | 67.0% | 50.8% | 40.8% | 38.7% | 55.9% | 72.1% | 56.6% | 56.9% | 8.7% |
| Recall avg    | 62.8% | 78.7% | 89.1% | 54.2% | 32.6% | 23.7% | 28.4% | 32.6% | 29.4% | 19.1% | 3.5% |

5.3.1. Feature weight learning rate $\tau$. From the experimental results shown in table 1 and figure 14, increasing $\tau$ within a certain range will improve the overall tracking performance, but too large $\tau$ will cause the P and R to decrease. From the principle analysis, adaptively adjusting the feature weight can weaken the negative influence of environment feature to obtain a better tracking effect. Therefore, the tracking effect can be improved by setting a certain feature weight learning rate.

5.3.2. Feature learning rate $\alpha$. From the experimental results shown in table 2 and figure 15, when $\alpha = 0$, P is higher, but R is lower. Properly increasing the learning rate can increase R, thereby improving the overall tracking level; when the learning rate is further increased, the overall tracking level will decrease.

On the whole, setting an appropriate learning rate can achieve a better tracking effect.

6. Conclusion

The original intention of CFAL is to realize adaptive learning of historical features by skillfully designing feature structures, so that the specified targets can be effectively tracked without labels and large sample sets. And the algorithm can have strong universality, for any image sequence of a given target, can achieve effective tracking.
The experimental results show that CFAL can track the specified target effectively by setting appropriate parameters under partial occlusion and morphological change. But on the other hand, it also reflects the deficiency of this algorithm: for different tracking scenarios, different parameters need to be set, and there is no strong universality.

Through experimental analysis, the influence of each parameter on the final tracking effect is understood to some extent. If we find the law through a large number of experiments, we can further set the adaptive adjustment of parameters and realize a more generalized algorithm.

7. References

[1] S. J. McKenna, Y. Raja and S. Gong 1999 Tracking colour objects using adaptive mixture models *Image and Vision Computing*
[2] Boyoon Jung and Gaurav S. Sukhatme Detecting Moving Objects using a Single Camera on a Mobile Robot in an Outdoor Environment *the 8th Conference on Intelligent Autonomous Systems*
[3] Ying Wu, Thomas S. Huang 2004 Robust Visual Tracking by Integrating Multiple Cues Based on Co-Inference Learning *International Journal of Computer Vision*
[4] B. Georgescu, D. Comaniciu, T. Han, et al. 2004 Multi-model component-based tracking using robust information fusion *Proceedings of the 2nd Workshop on Statistical Methods in Video Processing*
[5] Ganesh G. Patil and Rohitash K. Banyal 2020 A Dynamic Unconstrained Feature Matching Algorithm for Face Recognition *Journal of Advances in Information Technology*
[6] CHEN Li-xue and CHEN Zhao-jiong 2008 Image Retrieval Algorithms Based on Lab Space *Computer Engineering*
[7] XU Yuan, XU Xiaoliang, LI Cainian, JIANG Mei and ZHANG Jianguo 2016 Pedestrian Detection Combining with SVM Classifier and HOG Feature Extraction *Computer Engineering*
[8] Y. Chen, S. Li, X. Lin 2011 Fast hog feature computation based on CUDA *IEEE International Conference on Computer Science and Automation Engineering*
[9] Mizuno K, Terachi Y, Takagi K, et al. 2012 Architectural Study of HOG Feature Extraction Processor for Real-time Object Detection *IEEE Workshop on Signal Processing Systems (SiPS)*
[10] Bauer S, Kohler S, Doll K, et al. 2010 FPGA-GPU Architecture for Kernel SVM Pedestrian Detection *Proceedings of 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*
[11] LI Xiao-yan, ZHANG Zi-gang, ZHANG Yi-shi and Zhang Mi 2012 KL-divergence Based Feature Selection Algorithm with the Separate-class Strategy *Computer Science*
[12] P. KaewTrakulPong and R. Bowden 2003 A real time adaptive visual surveillance system for tracking low-resolution colour targets in dynamically changing scenes *Image and Vision Computing*
[13] ZHANG Bo Research on Visual Tracking Algorithm Based on Particle Filter

Acknowledgments
This research work here is supported by Undergraduate Innovation Research Program of Nankai University (Grant No. 202010055072).