MEAN-SHIFT TRACKING ALGORITHM WITH WEIGHT FUSION STRATEGY

Lingfeng Wang, Chunhong Pan, and Shiming Xiang

NLPR, Institute of Automation, Chinese Academy of Sciences
{lfwang,chpan,smxiang}@nlpr.ia.ac.cn

ABSTRACT

In this paper, we propose a new Mean-shift algorithm to tackle some tracking difficulties, such as background clutter and partial occlusion. First, we compare all Mean-shift-like tracking algorithms, and indicate that the main difference among them is weight calculation. Then, a new fusion strategy is proposed to unify all weight calculation methods into a framework. Based on this framework, we propose a novel weight calculation method, which takes the *candidate* model into consideration as well as incorporates the local *background*. Extensive experiments are conducted to evaluate the proposed approach. Comparative experimental results indicate that the tracking accuracy is improved as compared with the state-of-the-arts.

Index Terms— Mean-shift, Fusion strategy

1. INTRODUCTION

Real-time object tracking is a fundamental task for various applications, such as surveillance [1]. Many methods have been proposed to solve it. Among them, Mean-shift tracking algorithm is a popular one due to its simplicity and efficiency. The principle of Mean-shift tracking algorithm is to track the centroid of target by combining sample weights in a local neighborhood with a kernel function. Hence, two fundamental problems need to be addressed. The first problem is how to produce the sample weight in *search-region*, while the second one is how to select a appropriate kernel function [2]. In this work, we mainly focus on the first difficulty.

Previous Works: Mean-shift-like tracking algorithms, including the original Mean-shift tracking algorithm [3] and their improved variants, can be mainly classified into two groups, i.e., without-background and with-background.

Without-background methods, such as Cam-shift [4] and original Mean-shift [5, 3], only consider the target appearance. Ning et al. [6] pointed out background information is actually not used in [3]. The main limitation of these methods is that they are prone to local minima when some of target features appear in the background. The main reason is that they ignore local background information. In many tracking applications, a tracker only needs to separate the target from its local background. Hence, it is a very restrictive assumption if it only utilizes the *target* model but ignores its surroundings.

With-background tracking algorithms, such as online-selection [7], NBP Mean-shift [8], and CBWH Mean-shift [6], are proposed to decrease background interference in target representation. The strategy of these methods is to derive a simple representation of background features, which are then used to select the salient components from *target* model. Improved performances compared with standard Mean-shift method have been reported in these papers. Unfortunately, they both have their own shortcomings. For example, online-selection and NBP Mean-shift do not adopt the *candidate* model, while CBWH Mean-shift utilizes the background simply. The main reason is that no unified framework is proposed to combine them together as well as make them complement with each other.

Method: Motivated by previous works, an improved Mean-shift tracking algorithm, which relies on a novel weight calculation method, is proposed. The weight calculation method proposed in this work mainly depends on a fusion strategy, which unifies all weight calculation together into a framework. The contributions of this work are listed as follows:

1. We summarize all Mean-shift-like algorithms into a unified fusion framework on weight calculation. Original Mean-shift [3], online-selection [7], NBP Mean-shift [8], and CBWH Mean-shift [6] can be regarded as examples of our fusion strategy. Accordingly, it is very easy to add other terms on weight calculation to enhance the classical Mean-shift tracking algorithm.

2. Based on the fusion framework, we propose a new weight calculation method, which combines *candidate* model and local *background* together. Our weight calculation method holds the following two advantages. First, compared with the online-selection [8] and NBP Mean-shift [7], we incorporate the *candidate* model. Thus, our method can track the target more smooth. Second, compared with Mean-shift [3] and CBWH Mean-shift [6], our method represents the background as *foreground against background* formulation. Hence, our method is robust to the case when the initial target contains much background. Moreover, the computation cost of our method is low.

2. THE PROPOSED METHOD

2.1. Comparison of Mean-shift-like Algorithms

The strategy of tracking is to search a *candidate* that is most similar to *target* model between consecutive frames. Fig. 1 illustrates all types of regions in detail. To satisfy the low-computational cost imposed by real-time processing, all types of regions are usually represented by rectangles, and color features are used to describe them by calculating their color histograms. Histograms of *target*, *candidate*, and local *background* are denoted by $\mathbf{p}$, $\mathbf{q}(\mathbf{y})$, and $\mathbf{r}$, respectively, where $\mathbf{y}$ is the *candidate* location. Take the *target* histogram $\mathbf{p}$ as example, $\mathbf{p} = \{p_u\}_{u=1}^{\mathrm{B}}, \sum_{u=1}^{\mathrm{B}} p_u = 1$, where B is the bin number and u is bin index. A key issue in the Mean-shift-like tracking algorithms is the task of updating the current location $\mathbf{y}_{m+1}$ from the previous location $\mathbf{y}_m$ according to the mean shift iteration equation (note that kernels with Epanechnikov profile are recommended to be used for the calculation of histograms [3])

$$\mathbf{y}_{m+1} = \frac{\sum_{i=1}^{n} \mathbf{x}_i \omega_i}{\sum_{i=1}^{n} \omega_i}, \tag{1}$$

This work was supported by the Projects (Grant No. 60873161 and 60975037) of the National Natural Science Foundation of China.

Table 1. Comparison of the weight ω_i calculations by Mean-shift-like tracking algorithms.

Algorithm	Weight Calculation	Using Candidate	Using Background	Foreground against Background
Cam-shift [4]	p_u	No	No	No
Mean-shift [3]	$\sqrt{\frac{p_u}{q_u(\mathbf{y}_m)}}$	Yes	No	No
Online-selection [8]	$\log \frac{\max\{p_u,\epsilon\}}{\max\{r_u,\epsilon\}}$	No	Yes	Yes
NBP Mean-shift [7]	$\frac{\max\{p_u,\epsilon\}}{\max\{p_u+\lambda r_u,\epsilon\}}$	No	Yes	Yes
CBWH Mean-shift [6]	$\sqrt{\min\left\{\frac{r^\star}{r_u},1\right\}}\sqrt{\frac{p_u}{q_u(\mathbf{y}_m)}}$	Yes	Yes	No
Proposed Method	$\frac{\max\{p_u,\epsilon\}}{\max\{r_u,\epsilon\}}\sqrt{\frac{p_u}{q_u(\mathbf{y}_m)}}$	Yes	Yes	Yes

where $\{\mathbf{x}_i\}_{i=1}^n$ are pixels in the *candidate* region centered at $\mathbf{y}_m$, and $\{\omega_i\}_{i=1}^n$ are responding weights.

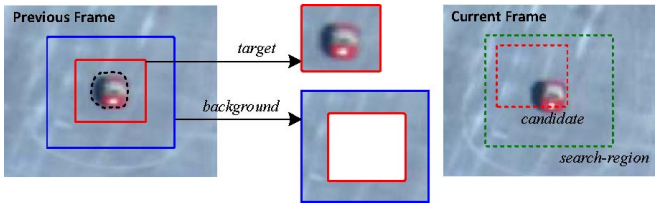

Fig. 1. All types of regions. In previous frame, the *target* and its local *background* are represented by red and blue rectangles, respectively. In current frame, the *candidate* and *search-region* are represented by red and green dashed rectangles, respectively.

The main difference of all Mean-shift-like tracking algorithms is the calculation of weight ω_i. Tab. 1 compares these algorithms in detail. Cam-shift [4] algorithm adopted *target histogram back-projection* technique, i.e., $\omega_i = p_u$, to calculate the weight. This algorithm performs well only when target appearance is homogeneous. Different from Cam-shift, classical Mean-shift algorithm improved it by dividing p_u with the *candidate* $q_u(\mathbf{y})$:

$$\omega_i = \sqrt{\frac{p_u}{q_u(\mathbf{y}_m)}}, \tag{2}$$

where $\sqrt{\frac{p_u}{q_u(\mathbf{y}_m)}}$ is named the *target against candidate* (**TAC**) term. Compared to Cam-shift, the tracking process of Mean-shift is more smooth, especially, when the tracked target is inhomogeneous. However, both two methods may fail if similar background surrounds the target. The main reason is that the local *background* is ignored by these methods. Hence, it is necessary to incorporate the local *background* to enhance the weight calculation. Online-selection algorithm [3] computed log likelihood ratio between *target* and *background* to form the weight:

$$\omega_i = \log \frac{\max\{p_u,\epsilon\}}{\max\{r_u,\epsilon\}}, \tag{3}$$

where ϵ is a small constant. Motivated by [3], NBP Mean-shift [7] incorporated the local *background* based on the Bayes' theorem:

$$\omega_i = \frac{\max\{p_u,\epsilon\}}{\max\{p_u+\lambda r_u,\epsilon\}} \approx \frac{1}{1+\lambda\frac{\max\{r_u,\epsilon\}}{\max\{p_u,\epsilon\}}} \approx \frac{\max\{p_u,\epsilon\}}{\max\{r_u,\epsilon\}}, \tag{4}$$

where λ is the priori ratio (see [7] for details). Both two methods utilize *background* as *target against background* (**TAB**) formulation, that is, $\frac{\max\{p_u,\epsilon\}}{\max\{r_u,\epsilon\}}$. From the **TAB** term, we can see that it emphasizes features that better distinguish from *target* to *background*. Hence, the interference by the similar background color can be removed. Unfortunately, neither of them considers the *candidate* $q_u(\mathbf{y})$. Hence, similar to Cam-shift, their tracking processes are unstable. CBWH Mean-shift algorithm [6] calculated weight by consider both local *background* r_u and *candidate* $q_u(\mathbf{y})$, given by

$$\omega_i = \sqrt{\min\left\{\frac{r^\star}{r_u},1\right\}}\sqrt{\frac{p_u}{q_u(\mathbf{y}_m)}}, \tag{5}$$

where $r^\star$ is the minimal non-zero value in $\{r_u\}_{u=1}^{\mathrm{B}}$. The main problem of this method is that *background* is not used as the **TAB** term. Thereby, it is hard to track the target if its appearance is homogeneous and if similar background color exists.

2.2. Fusion-based Weight Calculation

As interpreted in Subsection 2.1, each term has their own advantages and disadvantages on weight calculation. Such as, the **TAB** term can reduce the disturbance from the background, while the **TAC** term can make tracking process smooth.

Totally, all these terms can be unified into a fusion-based weight calculation framework. In this framework, each term is regarded as a individual observation z from the input image $\mathbf{I}$, or more accurately, a distribution $p(z|\mathbf{I})$ depend on $\mathbf{I}$. Weight is formulated as the joint distribution of all observations, given by

$$\omega_i = p(z^1, z^2, \ldots, z^c|\mathbf{I}), \tag{6}$$

where c is the observation number. Here, we further assume all observations are conditional independence to the input image, that is,

$$\omega_i = p(z^1, z^2, \ldots, z^c|\mathbf{I}) = \prod_{i=1}^{c} p(z^i|\mathbf{I}). \tag{7}$$

Multi-cue fusion is widely used in *Particle filter* tracking algorithms. In some of them, cues or observations, such as color, edge, motion, and saliency, are just fused as Eqn. 7. The main superior of using fusion strategy is that they can make them complement with each other on the tracking process, which just is our main motivation. The slight difference is that in Mean-shift observations are formulated as weights, while in *Particle filter* they are represented as similarity.

In this work, two observations are used, that is, $c = 2$. The first observation is the **TAB** term $p(z^1|\mathbf{I})$, while the second one is the **TAC** term $p(z^2|\mathbf{I})$, given by

$$\omega_i = p(z^1|\mathbf{I})p(z^2|\mathbf{I}) = \frac{\max\{p_u,\epsilon\}}{\max\{r_u,\epsilon\}}\sqrt{\frac{p_u}{q_u(\mathbf{y}_m)}}. \tag{8}$$

Online-selection [8] and NBP Mean-shift [7] methods only consider the **TAB** observation. That is, in their method, the **TAC** observation $p(z^2|\mathbf{I})$ is regarded as the Uniform distribution. Similarly, Mean-shift [3] considers the **TAB** observation $p(z^1|\mathbf{I})$ as the Uniform distribution. From the above description, we can conclude that these methods are special cases of ours.

2.3. Target and Background Updating

In order to cope with the changes of target appearance and background during tracking, histograms of $\mathbf{p}$ and $\mathbf{r}$ are updated frame-by-frame. To reduce the drifting probability, we use a slight model updating based on Bhattacharyya similarity. For *target* model, if the Bhattacharyya similarity between previous $\mathbf{p}(t)$ and current observed $\widehat{\mathbf{p}}(t+1)$, i.e., $\rho(\mathbf{p}(t), \widehat{\mathbf{p}}(t+1))$, is larger than τ (τ is experimentally set 0.8), the *target* $\mathbf{p}_{t+1}$ is updated as follows:

$$\mathbf{p}_{t+1} = (1 - \eta^{\mathbf{p}})\, \mathbf{p}(t) + \eta^{\mathbf{p}} \widehat{\mathbf{p}}(t+1). \tag{9}$$

Otherwise, the *target* $\mathbf{p}_{t+1}$ will not be updated. To avoid the excessive update of target model, $\eta^{\mathbf{p}}$ is set as a small value. The same procedure is used for *background* model, yet with different update ratio $\eta^{\mathbf{r}}$. In our implementation, we set $\eta^{\mathbf{p}} = 0.01$ and $\eta^{\mathbf{r}} = 0.1$, experimentally. Our tracking algorithm is summarized in Alg. 1.

Algorithm 1: Fusion-based Mean-shift Algorithm

Calculate target model $\mathbf{p}$ *and its local background* $\mathbf{r}$;
for $t \Leftarrow 1$ **to** frameNumber **do**
  $\mathbf{y}_1 =$ *previous position*;
  for $m \Leftarrow 1$ **to** maxIter $= 20$ **do**
    Calculate the target candidate $\mathbf{q}(\mathbf{y}_m)$;
    Calculate the weight $\{\omega_i\}_{i=1}^n$ *according to* Eqn. 8;
    Calculate new position $\mathbf{y}_{t+1}$ *of candidate* by Eqn. 1;
    Let $d = \|\mathbf{y}_{m+1} - \mathbf{y}_m\|$, $\mathbf{y}_{m+1} = \mathbf{y}_m$;
    if $d < \varepsilon$ *(*ε *is set* 0.1*)* **then**
      Update target and local background;
      break;
    end
  end
end

3. EXPERIMENTAL RESULTS

Extensive experiments have been conducted to evaluate the performance proposed algorithm, quantitatively and qualitatively. In all the experiments, the **RGB** color model is used as the feature space and it was quantized into $16 \times 16 \times 16$ bins. Quantitative evaluation is done with **F-score**, which measures the tracking accuracy by considering both the recall and the precision, which is defined as

$$\textbf{F-score} = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FN} + \mathrm{FP}} \tag{10}$$

where TP, FP, and FN are the true positives (true target pixels), false positives (number of background pixels that are masked as target), and false negatives (number of target pixels that are missed), respectively. Ground truth is created by manually selecting the image region considered by human as the best match of the object.

Two Difficulties: In the first experiment, we give two tracking results with some difficulties, such as background clutter (see up sequence of Fig. 2) and partial occlusion (see down sequence). The video sequences are provided by [9]. As shown in Fig. 2, despite these difficulties, our method still track the target well. Especially, the initialization of second example is not very accurate.

Fig. 2. Tracking results of our method with some difficulties, such as background clutter and partial occlusion.

Compare with State-of-the-arts: In the second experiment, two test sequences are selected to evaluate our method by comparing with several state-of-the-arts, i.e., Cam-shift [4], classical Mean-shift [3], and CBWH Mean-shift [6]. The video sequences are downloaded from the web-site [1]. In the first sequence, the target is surrounded by clutter background, while in the second one, the illumination condition is varied as target moves. Fig. 3 illustrates all the tracking traces in detail. As shown in this figure, the proposed algorithm can track targets smoothly, while others are fail at last. The numeric comparisons based on **F-score** are shown in Fig. 4. As a whole, **F-score** of our method is higher than others.

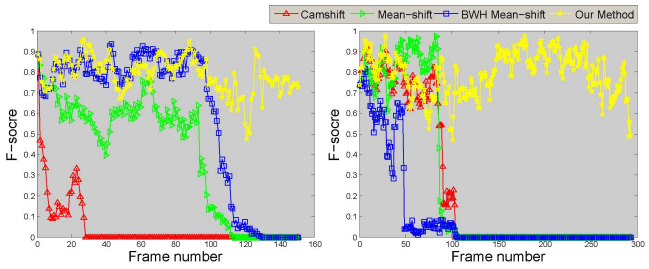

Fig. 4. Numeric comparison (by **F-score**s) of our method (in yellow) with several state-of-the-arts, i.e., Cam-shift (in red), Mean-shift (in green), and BWH Mean-shift (in blue).

Initialization Testing: The third tracking example gives tracking results of aerial footage of car, which is provided by [9]. This video sequence is utilized to test the robustness of the proposed method to inaccurate target initialization. As shown in Fig. 5, when the initial target is small, the result of our algorithm is as the same as that of the classical Mean-shift. However, when the initial target is large, that is, the target contains a lot background, our algorithm can still track the target while the classical one can not. **F-score** comparisons are shown in Fig. 6. As shown in this figure, compared to Mean-shift, our method holds higher **F-score**s. Moreover, Mean-shift algorithm performs well only when proper initialization is supplied, such as **Min initialization**. Accordingly, we can conclude that our algorithm is less sensitive to the quality of initialization than the classical Mean-shift. This tracking result indicates that the influence

[1] http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/

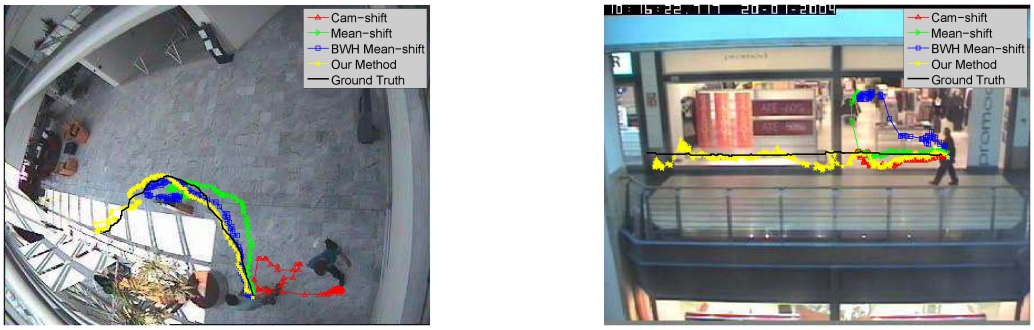

Fig. 3. Visual comparison of our method (in yellow) with several state-of-the-arts, i.e., Cam-shift (in red), Mean-shift (in green), and BWH Mean-shift (in blue). The black solid curves are the ground truth traces.

of background pixels on the *target* representation can be eliminated by the **TAB** term.

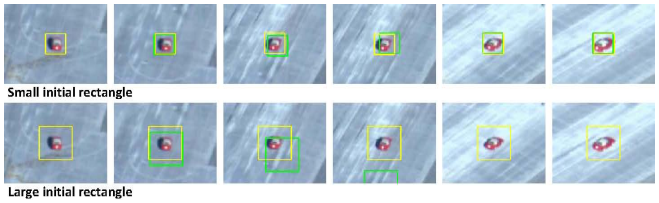

Fig. 5. Tracking results in the same scene with two different initializations. Tracking results of our method are shown in yellow, while those of Mean-shift [3] are shown in green.

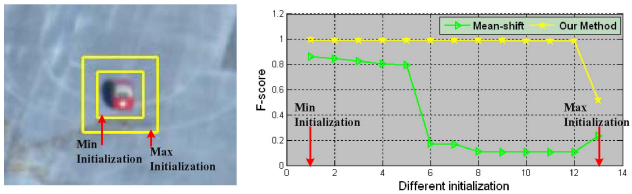

Fig. 6. Comparison with ground truth by **F-score**. Left: the different initializations, where only **Max** and **Min** initializations are drawn; Right: **F-score** comparisons.

Computation Cost: The speed of our algorithm is also measured in **fps** (frame per second). Here, the tested video sequences are the same as those used in **Compare with State-of-the-arts**. We use a standard PC with a 2.4 GHz processor and 1024 MB of memory, and choose the MATLAB in our experiments. Tab. 2 summarizes the average speeds and iteration numbers in detail. As shown in this table, the computation cost of our method is lower than most methods except the Cam-shift. The main reasons arise from two aspects. First, Cam-shift algorithm does not need to calculate the candidate model in each iteration, and the calculation candidate model is very time-consuming. Hence, Cam-shift algorithm can hold lowest computation cost even with higher number of iteration. Second, our method represents background as **FAB** term, therefore, the background features in search-region is suppressed, which causes that the number of iteration of our method is smaller than others. Hence, our algorithm is faster than Mean-shift and CBWH Mean-shift.

Table 2. Overview of comparison of computation cost. **Boldface**: best; Underline: second best.

Alg	Cam-shift	Mean-shift	CBWH Mean-shift	Our
Speed	**2.53**	1.67	1.72	1.95
Iter Num	7.34	5.32	3.09	**2.75**

4. CONCLUSION AND FURTHER WORKS

In this paper, we proposed a new tracking algorithm based on a new weight calculation strategy within Mean-shift framework. The new weight calculation method is derived from a fusion framework. Several examples are conducted to validate the proposed approach. Comparative experiments show its efficiency. In the further, we will combine take sophisticated filtering algorithms, such as *Kalman filter* and *Particle filter*, as an observation on weight calculation.

5. REFERENCES

[1] M. Greiffenhagen, D. Comaniciu, H. Niemann, and V. Ramesh, "Design, analysis and engineering of video monitoring systems: An approach and a case study," *IEEE Proceedings*, vol. 89, no. 10, pp. 1498–1517, 2001.

[2] Robert T. Collins, "Mean-shift blob tracking through scale space," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 234–240.

[3] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer, "Kernel-based object tracking," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[4] Lee J H, Lee W H, and Jeong D S, "Object-tracking method using back-projection of multiple color histogram models," vol. 2, pp. 668–671, 2003.

[5] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer, "Real-time tracking of non-rigid objects using mean shift," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2000, pp. 142–149.

[6] Jifeng Ning, Lei Zhang, David Zhang, and Chengke Wu, "Robust mean shift tracking with corrected background-weighted histogram," 2010.

[7] lingfeng Wang, Huaiyu Wu, and Chunhong Pan, "Mean-shift object tracking with a novel back-projection calculation method," in *Asia Conference on Computer Vision*, 2009, pp. 83–92.

[8] Robert Collins, Yanxi Liu, and Marius Leordeanu, "On-line selection of discriminative tracking features," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 1631–1643, 2005.

[9] Robert Collins, Xuhui Zhou, and Seng Keat Teh, "An open source tracking testbed and evaluation web site," in *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, January 2005.