A MEAN SHIFT ALGORITHM BASED ON MODIFIED PARZEN WINDOW FOR SMALL TARGET TRACKING

Jianjun Chen¹, Guocheng An², Suofei Zhang¹, Zhenyang Wu¹

¹School of Information Science and Engineering, Southeast University, Nanjing, China
²Intelligence Engineering Lab, Institute of Software Chinese Academy of Sciences, Beijing, China

ABSTRACT

This paper addresses the problem of small scale target tracking. The divided-by-zero problem in the weight computation of mean shift algorithm and its associated tracking interrupt problem are presented. To tackle these problems, the Parzen window density estimation method is modified to interpolate the histogram of the target candidate. Then the Kullback-Leibler distance is employed as a new similarity measure between the target model and the target candidate. Its corresponding weight computation and new location expressions are derived. On the basis of these works, we propose a new small target tracking algorithm using mean shift framework. The tracking experiments for real world video sequences show that the proposed algorithm can track the target successively and accurately. It can successfully track very small targets with only 6×12 pixels.

Index Terms—Small target tracking, Mean shift, Histogram interpolation, Parzen window, Similarity measure

1. INTRODUCTION

Small-size target tracking is commonly used in long distance outdoor visual surveillances and point light source tracking. Sometimes high dimensional motion parameters of the large-scale target are estimated by tracking different small components within the target independently in low dimensional parameter spaces [1]. Small targets contain only a few pixels which are much less than that of the whole image and the detailed information of the target appearance is usually unavailable. Therefore it is very difficult for traditional methods to track small targets. Some new approaches have been presented for this issue recently. An adaptive correlation-based small scale target tracker was developed using discriminant nonlinear filters that were robust to noise, background, and object distortions [2]. In [3], Kalman filter was used to predict the possible position in the next frame of the video stream and then mean shift was used to search in this neighboring range.

The color histogram-based mean shift is used widely in visual tracking because of its simplicity and robustness to target deformation, scale variance, and partial occlusion [4]. A combined histogram tracker was constructed by extending the histogram based tracking where a weighted combination of several different histograms can be used [5]. An adaptive binning color model based on mean shift clustering algorithm was presented in [6]. The color distribution of each cluster was modeled using independent component analysis (ICA) in their algorithm.

This paper proposes a novel Parzen window-based mean shift algorithm for small target tracking. Two principle problems in small target tracking, namely the tracking interrupt and target losing are analyzed. The Parzen window method is modified to interpolate the histogram of the target candidate. In order to improve the tracking accuracy, we take the Kullback-Leibler distance to evaluate the similarity between the target model and the target candidate. The tracking results of the proposed algorithm for very small target in conditions including illumination changes and partial occlusions is presented.

2. MAIN PROBLEMS IN MEAN SHIFT

2.1. Divided-by-zero problem

In mean shift tracking algorithm, the new recursive location \( y_i \) is computed according to the equation [4]

\[
y_i = \frac{\sum_{i=1}^{N} \omega_i x_i g \left( \frac{x_i - y_{i-1}}{h} \right)}{\sum_{i=1}^{N} \omega_i g \left( \frac{x_i - y_{i-1}}{h} \right)},
\]

where \( x_i \) is the pixel locations in the kernel centered at \( y_{i-1} \) with bandwidth \( h \), and \( g(\cdot) \) is the shadow of the kernel. The weight \( \omega_i \) is computed as

\[
\omega_i = \sum_{u=1}^{N} q_u / p_u(y_{i-1}) \delta \left[ b(x_i) - u \right],
\]

where \( q_u \) and \( p_u(y_{i-1}) \) are the histograms of the target model and the target candidate respectively; \( b(x_i) \) denotes the histogram bin index of location \( x_i \).

So each histogram bin of the target model \( q_u \) is divided by its corresponding histogram bin of the target candidate \( p_u(y_{i-1}) \) to compute the weight \( \omega_i \). As small scale target contains only a few pixels, most histogram bins of the target model and the target candidate have zero values. Consequently, there would be divided-by-zero problem in the weight computation if the denominator \( p_u \) equals to zero. The general numerical method is to set the weight as zero,
i.e. set \( \omega_i = 0 \) when \( b(x_i) = u \) and \( p_u = 0 \). In the tracking process, the color value of the tracked target usually changes with its motion and the environmental factors such as illumination change. In case where no non-zero bins of the target model and the target candidate are overlapped, i.e. \( q_u \cdot p_u = 0 \), for all \( u = 1 \ldots m \), all the weights \( \omega_i, i = 1 \ldots n \), will be zeros. Thus Eq.(1) will become zero-divided-by-zero problem, which leads to an oddity value. The tracking will be interrupted.

### 2.2. Target losing problem

The similarity measure function between the target model and the target candidate is very important in mean shift, because it determines the weight of each pixel directly. The Bhattacharyya coefficient is chosen as the similarity measure in current mean shift algorithm [4]-[6]

\[
\rho(y) \equiv \rho(p(y), q) = \frac{1}{\sum_{u=1}^{m} \sqrt{p_u(y)q_u}}.
\]

As is shown in the expression, the target model \( q_u \) plays the same role with the target candidate \( p_u \) in Bhattacharyya coefficient. The target model is fixed in the tracking (if it is not updated), whereas the target candidate changes with time. So if we treat \( q_u \) and \( p_u \) equally, the tracking accuracy will decrease. Some other difficulties of Bhattacharyya coefficient are that it is not very discriminative and its calculation is quadratic in the number of samples [7].

### 3. PARZEN WINDOW-BASED MEAN SHIFT

#### 3.1. Histogram interpolation

Let \( \{x_i^\ast\}_{i=1}^{n} \) be the pixel location in the target model centered at \( y^\ast \) with bandwidth \( h^\ast \), \( u = 1 \ldots m \) be the histogram bin index. Then the target model can be represented as

\[
q_u = c \sum_{i=1}^{n} \delta \left[ \frac{b(x^\ast_i) - u}{h^\ast} \right],
\]

where \( c \) is the normalization constant, \( \delta(\cdot) \) is the Kronecker delta function, and \( b(\cdot) \) is the histogram bin index function.

In the same way, we suppose \( \{x_i\}_{i=1}^{n} \) is the pixel location of the target candidate centered at \( y \) with bandwidth \( h \). The target candidate is computed as

\[
p_u(y) = c_h \sum_{i=1}^{n} \delta \left[ \frac{b(x_i) - u}{h} \right],
\]

where \( c_h \) is the normalization constant.

As is shown in section 2.1, the number of pixels in small scale target is so small that it will lead to tracking interrupt problem. Target template update can solve this problem to some degree. However, it faces some difficulties in practice, for example, the updating time is difficult to determine, the background pixels are easily introduced into the target template, etc. In this paper, we use a simple and effective histogram interpolation method to resolve this problem. The Parzen window approach is chosen to interpolate the histogram of the target candidate when it mismatches with the target model. Given a set of \( d \)-dimensional training samples \( U = \{u_1, u_2, \ldots, u_n\} \), the Parzen window density estimation is given by [8]

\[
p_u(u) = \frac{1}{nh^d} \sum_{i=1}^{n} \phi \left( \frac{u - u_i}{h^d} \right),
\]

where \( \phi(\cdot) \) is the nonnegative window function and \( h \) is the window bandwidth.

It can be seen that the Parzen window does not use the information of the bins surrounding the interpolated bin \( u \) sufficiently, thus it is unsuitable to be used directly in our histogram interpolation. We modify the original estimation method to get the new Parzen window-based histogram interpolation method which is given by

\[
p_u'(y) = \begin{cases} 
  p_u(y) & \text{if } p_u(y) \neq 0 \\
  \frac{1}{C} \sum_{v} \phi(u - v) p_u(y) & \text{if } p_u(y) = 0
\end{cases},
\]

where \( v \) is the index of the histogram bins that are included in window \( \phi(\cdot) \) and \( h \) is the interpolating window bandwidth. The normalization constant \( C \) is derived by imposing the condition \( \sum \phi(\cdot - v) / h \) is equal to 1, from which we can get \( C = \sum \phi(\cdot) \).

The histogram interpolating results of the proposed method show that the interpolated histogram does not change much from the original histogram; however the histogram bins in the target color region are interpolated with some small value. The divided-by-zero problem in the weight computation is resolved effectively.

#### 3.2. Similarity measure function

The similarity between the target model and the target candidate can be measured by some metric distances after their histograms are constructed. In this paper, we employ Kullback-Leibler distance as the similarity measure instead of the Bhattacharyya coefficient

\[
\rho(y) \equiv \rho(q_u, p_u'(y)) = \sum_{u=1}^{m} q_u \log \frac{q_u}{p_u'(y)}.
\]

The experimental results show that the Kullback-Leibler distance has larger extent than the Bhattacharyya coefficient when they are used to measure the distance of some distributions. This means that the Kullback-Leibler distance is more discriminative than the Bhattacharyya coefficient. As will be shown in the next section, the weight computation derived from the new similarity measure is simpler than the traditional one.

#### 3.3. Tracking algorithm

Using Taylor expansion around the value \( p_u'(y_0) \), \( \rho(y) \) can be approximated after the substitution of \( p_u'(y) \) as

\[
\rho(y) \approx \sum_{u=1}^{m} q_u \log \frac{q_u}{p_u'(y_0)} + 1 - c_h \sum_{i=1}^{n} \omega_i \left( \frac{y - x_i}{h} \right)^T,
\]

where weight \( \omega_i \) is computed by
\[
\omega_i = \sum_{i=1}^{n} \frac{q_{u_i}}{p'(y_0)} \delta \left[ b(x_i) - u \right].
\] (10)

As the first two terms of the distance (9) are independent of \(y\), its minimization value can be obtained by maximizing the third term. The third term represents the density estimation computed with kernel profile \(k(\cdot)\) at \(y\), with each pixel \(x_i\) being weighted by \(\omega_i\). Thus we can get the mode of \(\rho(y)\) by employing mean shift algorithm. \(\rho(y)\) recursively moves from the current location \(y_0\) to the new location \(y_1\) that is given by

\[
y_1 = \sum_{i=1}^{n} \omega_i x_i g \left( \frac{|x_i - y_0|}{h} \right) / \sum_{i=1}^{n} \omega_i g \left( \frac{|x_i - y_0|}{h} \right),
\] (11)

where \(g(x) = k(x)\), supposing that the derivative exists for all \(x \in [0, \infty)\). The main difference of the new algorithm with the traditional one is that the weight computation becomes more discriminative and simpler. If we choose the Epanechnikov profile, location \(y_1\) is reduced to

\[
y_1 = \sum_{i=1}^{n} \omega_i x_i / \sum_{i=1}^{n} \omega_i .
\] (12)

The complete algorithm using histogram interpolation and new similarity measure is summarized as follows:

**Algorithm 1** Parzen Window-based Mean Shift

Given: The color histogram of the target model \(\{q_u\}\), the target location \(y_0\) in the previous frame.

1. Initialize the target location \(y_0\) in the current frame, compute the kernel color histogram \(\{p_u(y_0)\} u=1, \ldots, m\).
2. If \(p_u(y_0)q_u = 0\), for all \(u=1, \ldots, m\), go to Step 3 to interpolate the histogram. Otherwise, go to Step 4 to continue tracking.
3. Interpolate the target candidate histogram \(\{p_u(y_0)\}\) according to Eq.(7) and get the new histogram \(\{p'(y_0)\}\).
4. Derive the weights \(\{\omega_i\}, i=1, \ldots, n^*\) according to Eq.(10).
5. Compute the new location \(y_1\) according to Eq.(12).
6. If \(\|y_1 - y_0\| < \varepsilon\), stop, set \(y_0 = y_1\), read the next frame and continue to track. Otherwise set \(y_0 = y_1\) and go to Step 1.

### 4. EXPERIMENTAL RESULTS

The proposed algorithm was evaluated in several sequences and only some representative results were presented here. The normalized \(rg\) color space quantized into 64×64 bins was chosen as the feature space. Gaussian kernel with a bandwidth \(h_N\) of 10 was used to interpolate the histogram. The maximal iteration number of mean shift and the stop threshold \(\varepsilon\) were set to 15 times and 1 pixel respectively.

The CAVIAR database [9] contains surveillance video sequences used for target behavior recognition. Sequence **Fight OneManDown** (FOMD) has 269 frames each with 384×288 pixels. The upper body of the fighter with size 14×16 pixels was tracked. As the target moved very fast and his coat color was similar with the ground color, the tracking task was rather challenging. Both the tracking results of the basic mean shift (BMS) and the proposed Parzen window-based mean shift (PWMS) algorithm were shown in Fig. 1. BMS lost the target at frame 14, 129, and 269, whereas PWMS could track the target constantly in the entire sequence. Fig. 2 gave the tracking errors of both algorithms where the ground truth location was calibrated manually. PWMS tracked more accurately than BMS before frame 130, after that their difference was not obvious.

There were 329 frames in sequence **OurSeq_1** each with 160×120 pixels. The target of size 6×12 pixels went through region with strong illumination change and partial occlusion. The results of some representative frames tracked by BMS and PWMS were shown in Fig.3. BMS interrupted tracking at frame 24, 116, 136, 172, 189, 192, 230, 276 and lost the target at frame 100, 107, 283, 305, 325 due to the low resolution of the target and the large illumination variation. PWMS could track the target successively in the whole sequence. Fig.4 gave the localization errors of both algorithms. PWMS tracked the target continuously and accurately with most errors within 5 pixels. As BMS interrupted and lost the target many times, its tracking error fluctuated abruptly.

---

**Fig. 1** Tracking results of sequence **FOMD**. The frame 14, 98, 130, 269 are shown. Top row: The results of BMS. Bottom row: The results of PWMS.
5. CONCLUSIONS

In this paper, we modify the Parzen window method to interpolate the histogram of the target candidate to resolve tracking interrupt problem. The Kullback-Leibler distance is employed as the similarity measure to improve the tracking accuracy which is associated with the target losing problem. The experimental results show that the proposed algorithm can successively track low resolution target with scale as small as 6×12 pixels.

6. ACKNOWLEDGEMENT

This work is supported by the National Nature Science Foundation of China (NSFC) under grant No. 60672094.

7. REFERENCES


