

Kernel-based visual tracking with continuous adaptive distribution

Risheng Han,^a Zhongliang Jing,^b and Yuanxiang Li^b

^aShanghai Jiao Tong University, School of Electronic, Information and Electrical Engineering, Shanghai, China, 200240
E-mail: hanrs@sjtu.edu.cn

^bShanghai Jiao Tong University, Institute of Aerospace Science & Technology, Shanghai, China, 200240

Abstract. The template updating problem of Kernel-based tracking (KBT) includes two aspects: target-scale update and target-model update. The proposed algorithm can update both tracking window's scale and target model by making use of continuous adaptive distribution. The ability of KBT can be complemented within its own framework with modest computation cost. The proposed tracking algorithm tries to get a balance between the stability of KBT and adaptability of CAMSHIFT for creating a robust tracker. © 2009 Society of Photo-Optical Instrumentation Engineers. [DOI: 10.1117/1.3125423]

Subject terms: mean shift; CAMSHIFT; histogram backprojection.

Paper 080994LR received Dec. 30, 2008; revised manuscript received Mar. 10, 2009; accepted for publication Mar. 13, 2009; published online May 1, 2009.

1 Introduction

Kernel-based tracking (KBT), which is also called “mean-shift” tracking, has attracted more and more attention because of its speediness and simplicity.¹ Generally, a visual target is tracked by extracting some kinds of target's feature, which can be called a template in the first frame, and then finding the image region that matches the template as closely as possible in remaining frames. One of the most critical challenges in creating a robust tracker is how to update the template. In KBT, the target's model can be regarded as tracking template and tracking window's scale is the only parameter. However, in traditional KBT, the target model is not updated during the whole tracking period, which leads to poor localization when the object changes its scale or appearance. Thus, the template updating problem of KBT includes two aspects: target-scale update and target-model update.

There is a simple solution to periodically update the target model according to a constant threshold on the similarity metric used in tracking. The model can be updated if the correlation of the model with the target is higher than a predefined threshold. However, using this simple solution always leads to tracking failure. Because each time the target model is updated, small errors are introduced in the location of the template. With each update, these errors accumulate and the tracking region steadily drifts away from the object. To deal with template-updating problem of KBT, some more effective updating methods have been proposed. The technique of “tracking through scale space” is proposed in Ref. 2. The method uses Lindeberg's theory

to select the best scale of tracking window. In Ref. 3, the multiple kernels-tracking method is proposed, and the dimensionality of the measurement space can be increased by using multiple kernels. In Ref. 4, the target model can be updated according to the distributions of the target intensity and the local standard deviation measures. In Ref. 5, the mean shift is modified to deal with dynamically changing color probability distributions derived from video frame sequences. The modified algorithm is called the continuously adaptive mean shift (CAMSHIFT). In CAMSHIFT, the continuous adaptive distribution image can be directly used to track the human face. However, compared to CAMSHIFT, KBT is more robust than CAMSHIFT in much other application, because the KBT uses a more static kernel histogram as the target model. If the CAMSHIFT and KBT are regarded as two extreme conditions of mean shift, then the proposed tracking algorithm tries to get a balance between the stability of KBT and adaptability of CAMSHIFT for creating a more robust tracker.

In KBT, both the target model and target candidate are characterized by a kernel-based histogram vector. This vector is a kind of image-probability distribution by weighting the histogram with a simple monotonically decreasing kernel profile. A kernel-based histogram can be denoted as follows:

$$\vec{q} = [q_u]_{u=1, \dots, m}, \quad \text{and } q_u = \frac{1}{C_h} \sum_{i=1}^n \text{Kernel}[X_i - c^0] \delta[b(X_i), u], \quad (1)$$

where $\{X_i\}_{i=1, \dots, n}$ are the pixel locations of the target, Kernel is a spatially weighting function centered at c^0 . The $\delta(\cdot)$ is the Kronecker delta function, $b(X_i)$ is a binning function that maps the color of $\{X_i\}_{i=1, \dots, n}$ into a histogram bin u with $u = \{1, \dots, m\}$, and C_h is a normalization term which makes $\sum_{u=1}^m q_u = 1$. The initial center of the target is denoted as c^0 . Similarly, the tracking features, called the “candidate model,” can be denoted as follows:

$$\vec{p}(c^k) = [p_u]_{u=1, \dots, m}, \quad (2)$$

where $p_u = 1 / C_h \sum_{i=1}^n \text{Kernel}(X_i - c^k) \delta[b(X_i), u]$, and the center of candidate region in sequent frame k is denoted as c^k .

Originally, KBT is derived from second-order Taylor expansion of Bhattacharyya coefficient, which is defined as follows:

$$B[\vec{p}(c^k), \vec{q}] = \sum_{u=1}^m \sqrt{p_u(c^k) q_u}. \quad (3)$$

The Bhattacharyya coefficient is a popular likelihood between two vectors. And the kernel-based method realizes target model tracking through maximizing the Bhattacharyya coefficient, as in

$$\Delta c^* = \arg \max_{\Delta c^k} B[\vec{q}, \vec{p}(c^k + \Delta c^k)]. \quad (4)$$

The KBT algorithm outputs the Δc^* , which determines the displacement of the object, the Δc^* is also called the “mean shift” vector, which can be computed using the following formulas, iteratively:

$$\Delta c^k = \frac{\sum_{i=1}^n \text{Kernel}(X_i - c^k) w(X_i) (X_i - c^k)}{\sum_{i=1}^n \text{Kernel}(X_i - c^k) w(X_i)}, \quad (5)$$

where

$$w(X_i) = \sqrt{\frac{q_u}{p_u(c^k)}}. \quad (6)$$

For finding the proper Δc^* in (4), an iterated computation is needed by computing the $w(X_i)$ using (6) and deriving Δc^k using (5). The Bhattacharyya coefficient is computed after the algorithm completion to evaluate the similarity between the target model and the chosen candidate model.

Continuously Adaptive Distribution

The primary difference between CAMSHIFT and mean shift is that CAMSHIFT uses a continuously adaptive probability distribution, which means the kernel-based histogram is recomputed for each frame. In KBT, the candidate model can also be recomputed and used to create the probability distribution image; thus, the candidate model can be regarded as a continuously adaptive distribution in this letter. And then, the target's probability distribution image can be created by histogram backprojection (HBP), which replaces the pixel values of the input image with the value of the corresponding bin of the histogram. Thus, if the candidate model can keep proper coherence with the target model in the tracking process, the target can be enhanced by HBP. In order to provide the range of probability values between 0 and 255, the histogram bin values are rescaled as follows:

$$\hat{p} = [\hat{p}_u]_{u=1 \dots m} \quad \text{and} \quad \hat{p}_u = \min\left(\frac{255}{\max(\hat{p})} p_u, 255\right)_{u=1, \dots, m}. \quad (7)$$

According to (7), the histogram bin p_u are rescaled from $[0, \max(\hat{p})]$ to the new range $[0, 255]$.

Zerth Moment of HBP Image

In CAMSHIFT, the scale of tracking window is determined by zeroth moment. After HBP, given $I(x, y)$ is the intensity of HBP image at (x, y) within the tracking window. The zeroth moment is computed as follows:

$$M_{00} = \sum_x \sum_y I(x, y). \quad (8)$$

And the tracking window's scale can be determined as follows:

$$S_x = \alpha \sqrt{M_{00}} \quad (9)$$

$$S_y = \beta S_x^k \quad (10)$$

where α is the scale proportion of tracking window's scale to target's scale. The β is the proportion of S_y to S_x . Because the size of the search window is rounded up to the current or next-greatest odd number, in practice, the tracking window is set larger than the target. In this paper, $\alpha = 1.1$.

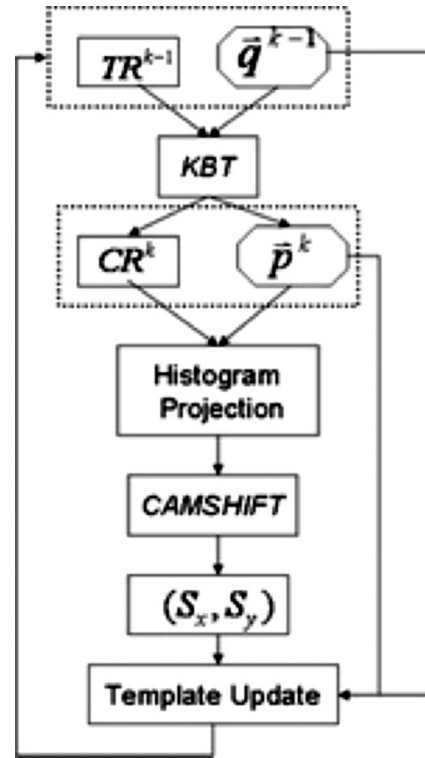


Fig. 1 Procedure of the proposed algorithm.

Visual Tracking with Continuous Adaptive Distribution

The proposed algorithm can be divided into three phases. First, estimate the tracked target's position using the KBT algorithm. Second, determine the tracking window's scale using the zeroth moment, which is computed by CAMSHIFT. Third, update tracking window's scale and target model. The proposed algorithm is illustrated by Fig. 1.

As Fig. 1 shows, the target region is denoted as TR^{k-1} and corresponding target model is denoted as \tilde{q}^{k-1} . The target's template information can be completely included in TR^{k-1} and \tilde{q}^{k-1} . After KBT is executed, the tracking window is moved to the candidate region, which is denoted as CR^k . In the candidate region, candidate model is denoted as \tilde{p}^k . On the basis of CR^k and \tilde{p}^k , a probability distribution image can be created, and then the CAMSHIFT can be executed to get the target's new scale information (S_x, S_y) . In the end, \tilde{p}^k , \tilde{q}^{k-1} , and (S_x, S_y) are used to update the template. Here, we give the proposed algorithm in detail. After the initial stage, the tracking starts from the $No.k$ ($k=2$) frame as follows:

Input: the target's initial center position (x_c^{k-1}, y_c^{k-1}) ; scale (S_x^{k-1}, S_y^{k-1}) in the previous frame.

Step 1: Determine tracking region: $TR^{k-1} = \text{RECT}(x_c^{k-1}, y_c^{k-1}, S_x^{k-1}, S_y^{k-1})$.

Step 2: Compute mean-shift vector $\Delta c^k = (\Delta x^k, \Delta y^k)$ using (5) and (6). Update candidate region: $CR^k = \text{RECT}(x_c^{k-1} + \Delta x^k, y_c^{k-1} + \Delta y^k, S_x^{k-1}, S_y^{k-1})$. Extract new candidate model \tilde{p}^k using (2) in CR^k .

Step 3: Transform CR^k by Histogram-Projection according to (7) and \tilde{p}^k .

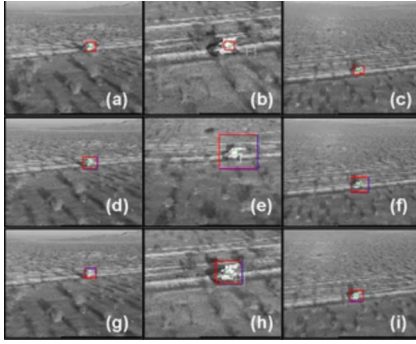


Fig. 2 Tracking results comparison: the proposed algorithm can update the tracking window's scale properly while KBT and CAMSHIFT cannot fit the target's varying scale properly.

Step 4: Compute the M_{00} using (8) iteratively until CAMSHIFT converges.

Step 5: Compute tracking window scale (S_x, S_y) using (9) and (10).

Step 6: According to (3), compute $B(q^{k-1}, p^k)$ with new scale (S_x, S_y) .

Output: If $B(q^{k-1}, p^{k-1}) < B(q^{k-1}, p^k)$

$$S_x^k = S_x; \quad S_y^k = S_y$$

$$\tilde{q}^k = B(\tilde{q}^{k-1}, \tilde{p}^k) * \tilde{p}^k + (1 - B(\tilde{q}^{k-1}, \tilde{p}^k)) * \tilde{q}^{k-1} .$$

Else

$$S_x^k = S_x^{k-1}; \quad S_y^k = S_y^{k-1}; \quad \tilde{q}^k = \tilde{q}^{k-1} .$$

End.

In the proposed algorithm, except for using the target's kernel histogram as the tracking feature, the target's continuously probability distribution image is also used. The zeroth moment of the HBP image is used to estimate the target scale based on CAMSHIFT. However, differences are that the target's position is computed by KBT in the tracking phase and the Bhattacharyya coefficient is used as an updating indicator and weighting factor in the updating phase. As the updating phase shows the target model has become a dynamic distribution that is adaptive to likelihood between current and previous distributions.

Results

In the test video sequence, the target's scale and appearance are changed constantly in a dynamic clutter background. As Figs. 2(a)–2(c) show, traditional KBT can not fit the target's varying scale in tracking process. Figures 2(d)–2(f) show that the tracking performance of CAMSHIFT is also inefficient. Because of the clutter background, CAMSHIFT always gets a tracking window that is too large. The tracking results of the proposed algorithm are shown in the Figs. 2(g)–2(i); the tracking window's scale can be properly updated and the target can be discriminated from a clutter background effectively.

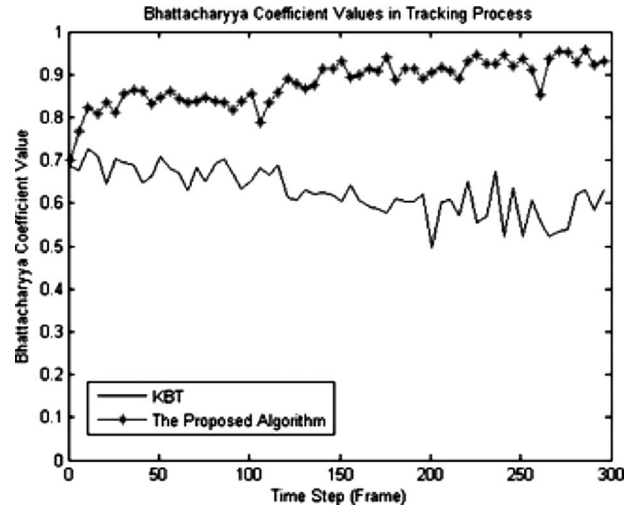


Fig. 3 Bhattacharyya coefficient values produced by KBT, and the proposed algorithm.

In Fig. 3, the similarity between target model and candidate model is also illustrated. Comparing to traditional KBT, the proposed algorithm can keep better coherence with the tracking template, which means the target model has been updated properly.

Conclusion and Discussion

There are two major contributions in this letter. First, a new scale updating method is proposed. By using target's continuously adaptive distributions image, the proposed algorithm can update the tracking window's scale. Second, by integrating the continuous updating method into KBT, the target model can also be updated within the framework of mean shift. Experiments show the proposed method can keep the tracking target even when the target scale and appearance are changed constantly in a clutter background. On the basis of the proposed algorithm, further extensions and improvements could be made by combining multiple color or texture spaces.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 60775022).

References

1. D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.* **25**, 564–577 (2003).
2. R. Collins, "Mean-shift blob tracking through scale space," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 234–240 IEEE Computer Society, Washington, DC (2003).
3. G. D. Hager, M. Dewan, and C. V. Stewart, "Multiple kernel tracking with SSD," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 790–797, IEEE Computer Society, Washington, DC (2004).
4. Y. Alper, S. Khurram, and S. Mubarak, "Target tracking in airborne forward looking infrared imagery," *Image Vis. Comput.* **21**, 623–635 (2003).
5. G. R. Bradski, "Computer vision face tracking as a component of a perceptual user interface," in *Workshop on Applications of Computer Vision*, pp. 214–219 (1998).