Kernel based visual tracking with scale invariant features

Risheng Han (韩日升)[1], Zhongliang Jing (敬忠良)[2], and Yuanxiang Li (李元祥)[2]

[1]*School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240*

[2]*Institute of Aerospace Science & Technology, Shanghai Jiao Tong University, Shanghai 200240*

Received August 14, 2007

The kernel based tracking has two disadvantages: the tracking window size cannot be adjusted efficiently, and the kernel based color distribution may not have enough ability to discriminate object from clutter background. For boosting up the feature's discriminating ability, both scale invariant features and kernel based color distribution features are used as descriptors of tracked object. The proposed algorithm can keep tracking object of varying scales even when the surrounding background is similar to the object's appearance.

OCIS codes: 330.7310, 330.0330, 100.0100, 070.5010.

The efficiency of visual tracking method depends on two factors: tracking features and tracking algorithm. Various popular visual cues have been studied, such as contour, appearance, corner points, histograms and so on. However, none of them is robust individually, so multiple cues are combined or selected in tracking process recently[1]. The major difficulty is how to efficiently integrate multiple cues with spatial constraints and temporal dynamics in a principled way.

In this letter, we propose an algorithm that makes use of both scale space features and color space features to realize robust visual tracking. After briefly reviewing the kernel based tracking (KBT)[2] and scale invariant features transform (SIFT) algorithms[3], a facile SIFT features matching algorithm is proposed. To overcome the unstableness of SIFT features matching algorithm and disadvantages of KBT algorithm, the kernel based scale invariant features tracking algorithm is proposed. Efficiency of the proposed algorithm is shown in experiments.

KBT algorithm[2] has attracted more and more attentions in recent years because of its speediness and simplicity. In KBT, the tracking template called target model can be denoted as

$$\vec{q} = [q_u]_{u=1,\cdots,m}\,, \quad \text{and}$$

$$q_u = \frac{1}{C_h}\sum_{i=1}^{n} \text{Kernel}(X_i - c^0)\cdot\delta(b(X_i),u), \tag{1}$$

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

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

where $p_u = \frac{1}{C_h}\sum\limits_{i=1}^{n}\text{Kernel}(X_i - c^k)\cdot\delta(b(X_i),u)$, and the center of candidate region in the kth frame is denoted as c^k. KBT is also called "mean shift" which is derived from second order Taylor expansion of Bhattacharyya coefficient denoted in

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

And the "mean shift" realizes target model tracking through maximizing Bhattacharyya coefficient, like

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

The KBT or "mean shift" outputs Δc^*, which determines the displacement of the tracked object,

$$\Delta c^k = \frac{\sum\limits_{i=1}^{n}\text{Kernel}(X_i - c^k)\cdot w(X_i)\cdot(X_i - c^k)}{\sum\limits_{i=1}^{n}\text{Kernel}(X_i - c^k)\cdot w(X_i)}, \tag{5}$$

where

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

To find the proper Δc^* in Eq. (4), an iterated computation is needed by computing $w(X_i)$ using Eq. (6) and deriving Δc^k using Eq. (5). To adjust the scale of tracking window, the scale is detected by calculating the Bhattacharyya coefficient for three different scales (larger, same and smaller by 5%) and choosing the scale that gives the highest similarity to the target model. But the scale of the tracking windows cannot always keep up with the object scale changes and lead to poor localization[4]. So, the first disadvantage of KBT is that the tracking window's size cannot be adjusted efficiently. In addition, the feature used by KBT always has not enough ability to discriminate object from clutter background. For solving the first problem, the technique of "tracking through scale space" was proposed[5]. The method uses Lindeberg's theory to select the best scale of tracking window size. For boosting up the feature's discriminating ability, scale invariant features are used as tracking features in our study.

Scale invariant features produced by SIFT have high probability to find the exact match under certain extent of illumination changes and affine transformation[3], so it is reasonable to use SIFT features in visual tracking. The SIFT algorithm outputs a set of features, and every feature consists of four items,

$$F_i = \{P_i, S_i, O_i, D_i \,|i = 1, \cdots, N\}, \qquad (7)$$

where the subscript i is serial number of SIFT feature. $P_i = \langle x_{pi}, y_{pi} \rangle$ is the ith SIFT feature's position; S_i, O_i and D_i denote the ith SIFT feature's direction, scale and descriptor, respectively. It should be noted that D_i is a grid of gradient orientation histogram, which is invariant to scale, rotation, translation and varying illumination. Let us give a facile SIFT features matching algorithm firstly.

It is important to note that features used for tracking only need to be locally discriminative, which means that the object only needs to be clearly separable from its immediate surroundings. In our study, both object and neighborhood background are considered as one tracking region which can be modeled as local SIFT detection field,

$$\text{SIFT\_LDF}(x_c^{k-1}, y_c^{k-1}) = \text{RECT}(\alpha \cdot S_x^{k-1}, \beta \cdot S_y^{k-1}), \quad (8)$$

where $\langle x_c^{k-1}, y_c^{k-1} \rangle$ is the center of the tracked object given by tracking result in previous frame; $\{S_x^{k-1}, S_y^{k-1}\}$ represents the scale of tracked object in horizontal and vertical directions; α and β are two constants which determine the proportion between object region and neighborhood background (typically, $\alpha = \beta = 1.5$). In the SIFT_LDF region, SIFT features can be divided into two subsets according to SIFT features' position item (denoted as D_i in Eq. (7)). One subset includes SIFT features which belong to the tracked object, and the other subset includes those SIFT features which belong to neighborhood background. In the initial frame, they are denoted as Init_Obj_Subset and Init_Bg_Subset respectively. According to the proportion of the object and neighborhood background, both subsets can be easily determined as

$$\text{Init\_Obj\_Subset}$$
$$= \left\{ \begin{array}{l} \langle x_{pi}^{k-1}, y_{pi}^{k-1} \rangle \in \text{SIFT\_LDF}| \\ x_{pi}^{k-1} \in [(x_c^{k-1} - S_x^{k-1}), (x_c^{k-1} + S_x^{k-1})] \wedge \\ y_{pi}^{k-1} \in [(y_c^{k-1} - S_y^{k-1}), (y_c^{k-1} + S_y^{k-1})] \end{array} \right\}, \quad (9)$$

$$\text{Init\_Bg\_Subset}$$
$$= \left\{ \begin{array}{l} \langle x_{pi}^{k-1}, y_{pi}^{k-1} \rangle \in \text{SIFT\_LDF}| \\ \langle x_{pi}^{k-1}, y_{pi}^{k-1} \rangle \notin \text{Init\_Obj\_Subset} \end{array} \right\}, \qquad (10)$$

where $k = 1$. Although the Init_Obj_Subset is determined only by SIFT features' position item, the other three items of each SIFT feature can also be determined simultaneously. In fact, the Init_Obj_Subset region contains the object's tracking template. Traditional SIFT features match is to find the indices of the nearest neighbors of the given descriptors in the specified database by using Euclidean distance[3]. However, considering speediness, Bhattacharyya coefficient is more suitable than Euclidean distance in visual tracking. The likelihood of descriptors in two continuous frames is defined as

$$B(D(P_i^{k-1}), D(P_i^k)) = \sqrt{D(P_i^{k-1}) \cdot D(P_i^k)}, \qquad (11)$$

where $D(P_i^{k-1})$ denotes SIFT feature's descriptor at position P_i^{k-1} in previous frame, $D(P_i^k)$ denotes the corresponding SIFT feature's descriptor at position P_i^k in current frame. In addition, the proposed matching algorithm should consider video object's dynamic motion constraint which is defined as SIFT motion constraint as follows,

$$\text{SIFT\_MC}(x_{pi}^{k-1}, y_{pi}^{k-1})$$
$$= \left\{ \begin{array}{l} \forall \langle x_{pi}^k, y_{pi}^k \rangle \in \text{SIFT\_LDF}| \\ x_{pi}^k \in [x_{pi}^{k-1} - \text{MC}_x, x_{pi}^{k-1} + \text{MC}_x] \wedge \\ y_{pi}^k \in [y_{pi}^{k-1} - \text{MC}_y, y_{pi}^{k-1} + \text{MC}_y] \end{array} \right\}, \quad (12)$$

where MC_x and MC_y are the SIFT feature's motion constraint parameters in both horizontal and vertical directions (typically, $\text{MC}_x = \text{MC}_y = 3$). Only the matched SIFT features which satisfy the definition of Eq. (12) can be added into matched set denoted as Match_Obj_Subset. And the neighborhood background is redefined as Bg_Subset. The two subsets are defined as follows,

$$\text{Match\_Obj\_Subset}$$
$$= \left\{ \begin{array}{l} \langle x_{pi}^k, y_{pi}^k \rangle \in \text{SIFT\_LDF}| \\ B(D(P_i^{k-1}), D(P_i^k)) > \text{Th} \wedge \\ \langle x_{pi}^k, y_{pi}^k \rangle \in \text{SIFT\_MC}(x_{pi}^{k-1}, y_{pi}^{k-1}) \end{array} \right\}, \quad (13)$$

$$\text{Bg\_Subset}$$
$$= \left\{ \begin{array}{l} \langle x_{pi}^k, y_{pi}^k \rangle \in \text{SIFT\_LDF}| \\ \langle x_{pi}^k, y_{pi}^k \rangle \notin \text{Match\_Obj\_Subset} \end{array} \right\}, \qquad (14)$$

where $P_i^{k-1} = \langle x_{pi}^{k-1}, y_{pi}^{k-1} \rangle$ and $P_i^k = \langle x_{pi}^k, y_{pi}^k \rangle$; Th is threshold to judge whether the corresponding SIFT feature matches its template descriptor or not (typically, $\text{Th} = 0.85$).

To determine the SIFT_LDF which will be used by next frame, the parameters $\langle x_c^{k-1}, y_c^{k-1} \rangle$ and $\{S_x^{k-1}, S_y^{k-1}\}$ should be updated as $\langle x_c^k, y_c^k \rangle$ and $\{S_x^k, S_y^k\}$, which can be determined by using mean position and dispersion of matched SIFT features respectively as follows,

$$\langle x_c^k, y_c^k \rangle$$
$$= \left\langle \frac{\sum\limits_{x_{pi}^k \in \text{Match\_Obj\_Subset}} x_{pi}^k}{N}, \frac{\sum\limits_{y_{pi}^k \in \text{Match\_Obj\_Subset}} y_{pi}^k}{N} \right\rangle, (15)$$

where N is the number of elements in the Match_Obj_Subset

$$\{S_x^k, S_y^k\} = \{\max |x_c^k - x_{pi}^k|, \max |y_c^k - y_{pi}^k|\}, \qquad (16)$$

where $\langle x_{pi}^k, y_{pi}^k \rangle \in \text{Match\_Obj\_Subset}$.

SIFT features are usually unstable in practical tracking process, which makes the number of matched SIFT features become smaller and smaller. Then, the SIFT_LDF cannot be determined correctly, which leads both SIFTS algorithm and matching algorithm to fail. To cope with the problem, KBT algorithm is integrated into above matching algorithm.

When tracking starts, the target's initial position $\langle x_c^{k-1}, y_c^{k-1}\rangle$ and tracking window's scale $\{S_x^{k-1}, S_y^{k-1}\}$ are input into the proposed algorithm. To determine tracking templates in both scale feature space and color feature space, three steps are needed in the initial stage. Firstly, running SIFT algorithm in SIFT_LDF(x_c^{k-1}, y_c^{k-1}). Secondly, determining the Init_Obj_Subset using Eq. (9). Thirdly, extracting target model using Eq. (1), where $c^0 = \langle x_c^{k-1}, y_c^{k-1}\rangle$, and the scale of tracking window is determined by $\{S_x^{k-1}, S_y^{k-1}\}$ $(k = 1)$.

After initial stage, the tracking starts from the kth $(k = 2)$ frame as follows.

```
Input: the target's initial center position $\langle x_c^{k-1}, y_c^{k-1}\rangle$ and scale $\{S_x^{k-1}, S_y^{k-1}\}$ in the previous frame.
Step1: Run SIFT algorithm in SIFT_LDF$(x_c^{k-1}, y_c^{k-1})$ in the $k$th frame.
Step2: Extract candidate model of KBT using Eq. (2), where $c^k = \langle x_c^{k-1}, y_c^{k-1}\rangle$; scale of tracking window is determined by $\{S_x^{k-1}, S_y^{k-1}\}$.
Step3: Produce Match_Obj_Subset using Eq. (13).
Step4: Compute $\langle x_c^k, y_c^k\rangle$ and $\{S_x^k, S_y^k\}$ using Eqs. (15) and (16).
Step5: Compute $B(\vec{q}, \vec{p}(\langle x_c^k, y_c^k\rangle))$ with scale $\{S_x^k, S_y^k\}$.
Step6: Compute $\Delta c^*$ by running Eqs. (5) and (6) iteratively until coverage of KBT.
Output: If $B(\vec{q}, \vec{p}(\langle x_c^k, y_c^k\rangle)) < B(\vec{q}, \vec{p}(c^k + \Delta c^*))$
            SIFT_LDF$(c^{k-1} + \Delta c^*) = \mathrm{rect}(\alpha \cdot S_x^{k-1}, \beta \cdot S_y^{k-1})$
        Else
            SIFT_LDF$(x_c^k, y_c^k) = \mathrm{rect}(\alpha \cdot S_x^k, \beta \cdot S_y^k)$
            Update the candidate model using Eq. (2).
        End
```

In the proposed algorithm, Bhattacharyya coefficient is used as an indicator that judges whether the matching result is stable or not. On the one hand, once the matching result becomes unstable, SIFT_LDF can be maintained with the help of KBT algorithm. On the other hand, the two disadvantages of KBT can also be overcome by using SIFT features matching result when the matching algorithm is stable. In detail, tracking window's scale can be updated by $\{S_x^k, S_y^k\}$, and SIFT features can discriminate the tracked object from clutter background.

The vehicle sequence of PETS2001 is used as test video. In this sequence, the vehicle's scale is varying and its appearance is similar to surrounding background, so KBT is inefficient. As shown in Figs. 1(a) and (b), KBT cannot keep tracking the vehicle because the vehicle's color appearance is similar to the highway's surface and fence. Figures 1(c) and (d) show that the tracking performance is also inefficient because of SIFT features' unstableness. These matched features are labeled as white dots. The tracking effect of the proposed algorithm is shown in the Figs. 1(e) and (f), the tracking window's scale can be properly updated and the vehicle can be discriminated from clutter background. Figure 2 shows the

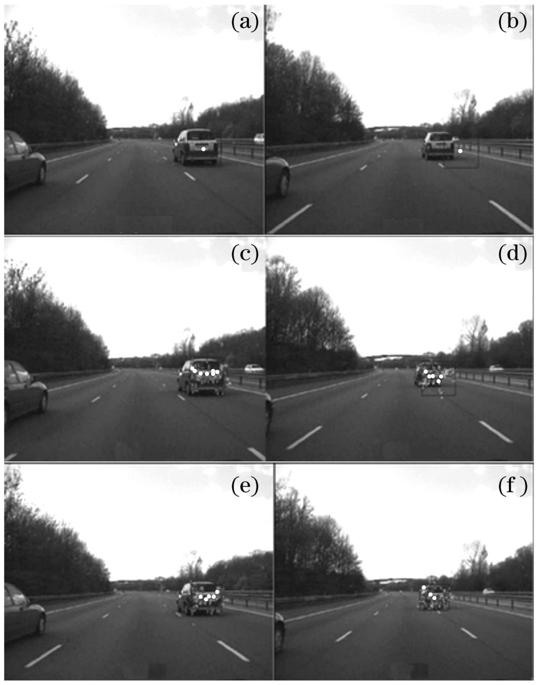

Fig. 1. Tracking results with (a,b) KBT, (c,d) SIFT features matching, and (e,f) the proposed tracking algorithm.

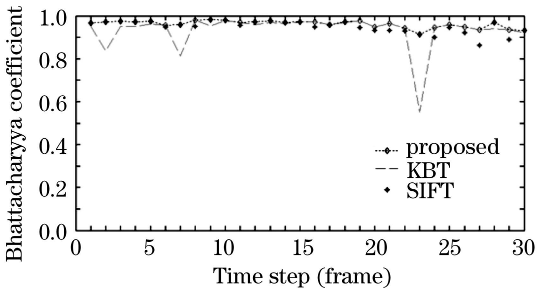

Fig. 2. Bhattacharyya coefficient values produced by KBT, SIFT features matching, and the proposed algorithm in tracking process.

Bhattacharyya coefficients produced by KBT, SIFT matching algorithm, and the proposed tracking algorithm. It is shown that the proposed algorithm can switch between KBT and SIFT features matching to get more accurate and robust tracking effects.

In the field of visual tracking, condensation tracking[6] and KBT might be the two most famous tracking methods in recent years. KBT is the representation of "data driven" tracking method, and the condensation tracking is the representation of "model driven" tracking method. From the viewpoint of computation cost, the "data driven" tracking method generally needs far less computation resource than "model driven" tracking method. The proposed algorithm can be regarded as a kind of "data driven" tracking method. So its computation cost is lower than most "model driven" visual tracking methods if the same features are used. On the other hand, it is obvious that the proposed algorithm's computation cost is higher than the original KBT because both scale space features and color space features are used. However, the added computation cost is worth its salt because tracking feature's discriminating ability can be boosted up and the shortages of KBT can be overcome. In addition, it should be noted that the original SIFT matching algorithm is fast enough for real-time applications even when running on a whole image. For example, it has been mentioned that the SIFT features matching can be done in less than a second even when matching an image to a large database[3]. In the proposed algorithm, the SIFT matching algorithm only needs to be executed within the area of SIFT_LDF, and the size of SIFT_LDF is far smaller than the size of whole image, so the computation cost can be more reduced. In short, the proposed algorithm only needs modest computation resource compared with other visual tracking algorithm.

In conclusion, the proposed algorithm makes KBT and the SIFT features matching become complementary in visual tracking. Experiments show how the proposed algorithm adapts to changing scale of tracked object. The tracker's efficiency can be guaranteed, even when the surrounding background is similar to the tracked object. In addition, an improving way is to estimate attitude parameters of tracked object in real time, because the stableness of object's SIFT features can be maintained by the proposed algorithm with modest requirement of computation resource.

This work was supported by the National Natural Science Foundation of China under Grant No. 60775022. R. Han's e-mail address is hanrs@sjtu.edu.cn.

References

1. A. Li, Z. Jing, and S. Hu, Chin. Opt. Lett. **3,** 326 (2005).
2. D. Comaniciu, V. Ramesh, and P. Meer, IEEE Trans. Pattern Anal. Mach. Intell. **25,** 564 (2003).
3. D. G. Lowe, Int. J. Computer Vision **60,** 91 (2004).
4. N. Peng, J. Yang, and Z. Liu, Opt. Eng. **44,** 7 (2005).
5. R. Collins, in *Proceedings of 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* II-234 (2003).
6. M. Isard and A. Blake, Int. J. Computer Vision **29,** 5 (1998).