Robust Multi-scale Prior L-K Tracking Based on Local Features

Kang SUN¹,* , Yue-wen GUO² and Jing ZHANG¹

¹School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo, China
²Department of State-owned Assets Management, Shanxi Agricultural University, Taigu, China

*Corresponding author

Keywords: Inverse Compositional, Multi-scale prior, Local feature, L-K Tracking.

Abstract. Due to the limitation of algorithm principle, inverse compositional L-K algorithm always fails to track fast moving targets. An improved tracking method based on multi-scale motion prior information is proposed in this paper. The multi-scale priori error Jacobi matrix is calculated offline by artificially constructed multi-scale disturbance. The target searching strategy based on multi-scale stratification can greatly increase the convergence range of the algorithm. At the same time, in order to improve the stability of global feature tracking in the case of local occlusion, the pyramid iterative L-K tracking method based on local feature is employed. Experimental results demonstrate outstanding stability of our method for fast moving targets. And by taking advantage of local gray information around feature points, our algorithm can still effectively track the target in the presence of partial occlusion.

Introduction

The mission of visual tracking is to estimate the location or area of object in video sequence in the case of detecting the initial position of the target. It is the basis of advanced applications such as public safety monitoring, intelligent vehicle and human-computer interaction, etc.¹⁻⁴

Since Lucas and Kanade proposed an image registration algorithm based on affine transformation, Lucas-Kanade algorithm has been widely applied in image stitching, motion estimation and face recognition.⁵ However, it suffers from inefficiency because of the need of recalculating and updating Hessian matrix at each iteration. Many scholars have made effective improvements.⁶⁻¹¹ Shum et al. proposed forward compositional algorithm (FC), which aimed at reducing the amount of computation required to update parameters during iteration. Dellaert et al. swap the roles between the current image and the template image to avoid recalculating Hessian matrices for each iteration, which greatly improves the efficiency of the algorithm. The idea was subsequently refined by Baker et al. to develop an efficient Inverse Compositional (IC) algorithm for real-time target tracking. Due to the principle limitation, the IC algorithm only can stably track the slowly moving target and usually fails to track high dynamic target.

In this paper, multi-scale motion prior information is employed to deal with this problem, in which the tracking problem is divided into two phases: offline training and online tracking. The multi-scale priori error Jacobian matrix acquired in offline training phase is used to solve real-time tracking stability under high dynamic environment. This method is essentially a global feature based tracking method, which uses the global dense or sparse gray information of the target template as a whole feature to describe the target, and it is difficult to achieve success when the target suffers from partial occlusion. Then, the pyramid iterative L-K tracking method based on local feature is applied to achieve more robustness.

Multi-scale Prior Tracking

Given the hypothesis of constant illumination during tracking process, tracking problem can be transformed into the optimization problem of geometry transform parameter $\mu(t)$.
\[ O(\mu(t)) = \left\| I(f(x; \mu(t)), t) - I(x, t_0) \right\|^2 \]  
where \( f(x; \mu(t)) \) is geometric distortion caused by the movement, \( \mu(t) = [\mu_1(t), \mu_2(t), \ldots, \mu_n(t)]^T \) is the transformation parameters, \( I(x, t_0) \) is the template.

By IC algorithm principle, the transformation parameter increment \( \Delta \mu(t) \) at time \( t \) can be obtain as the following least squares solution:

\[ \Delta \mu(t + \tau) = A(t_0)(I(x, t_0) - I(f(x; \mu(t)), t + \tau)) \]  
where \( A(t_0) = (R^T(x, t_0)R(x, t_0))^{-1}R^T(x, t_0) \) is called error Jacobian matrix, \( R(x, t_0) = [I_{\mu_1}(x, t_0), \ldots, I_{\mu_n}(x, t_0)]_{N \times n} \) is the Jacobian matrix of \( R \) with respect to \( \mu \) at time \( t_0 \).

It is obvious that \( A(t_0) \) is only related to the template image at time \( t_0 \), so it can be pre-computed before tracking without any update during the tracking process, which greatly reduces the amount of computation burden during tracking stage. Therefore, inspired by off-line training ideas, we could simulate arbitrary motion between two adjacent frames through artificially setting parameters \( \Delta \mu \), which we called prior motion. As the transformation parameters \( \Delta \mu \) is known, the tracking error corresponding this prior motion can be calculated as \( e(t + \tau) = I(x, t_0) - I(f(x; \mu(t)), t + \tau) \). Take \( \Delta \mu \) and \( e \) into (2), we can obtain the error Jacobian matrix corresponding to this prior motion, called priori error Jacobian matrix.

Replacing \( A(t_0) \) by \( A \), \( e = [e_1, e_2, \ldots, e_n]^T \) by \( e(t + \tau) \), \( \Delta \mu(t + \tau) \) by \( \Delta \mu = [\delta \mu_1, \delta \mu_2, \ldots, \delta \mu_n]^T \). Eq. (2) can be rewritten as.

\[ A \cdot e = \Delta \mu \]  

Figure 1. Prior motion distortion parameterized by four corner points.

In the training process, an artificially ‘disturbance’ \( \Delta \mu \) is firstly given to simulate the target template deformation. This prior disturbance can be achieved by randomly moving template vertex as shown in Fig.1. Here the ‘disturbance’ within the same amplitude range can be repeated \( N_p (N_p >> N) \) times randomly. Then, we can get the following equations:

\[ \begin{bmatrix} a_{i1} & \cdots & a_{ij} & \cdots & a_{ik} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{i1} & \cdots & a_{ij} & \cdots & a_{ik} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{i1} & \cdots & a_{ij} & \cdots & a_{ik} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} = \begin{bmatrix} \delta \mu_1 \\ \delta \mu_2 \\ \vdots \\ \delta \mu_n \end{bmatrix} \]  

From Eq.(4), we can calculate least squares solution of \( A \),

\[ O(A) = \sum_{k=1}^{N_p} (\Delta \mu_k^k - A \cdot e_k^k) \]  
where \( k \in [1, N_p] \), \( \Delta \mu_k^k = [\delta \mu_1^k, \delta \mu_2^k, \ldots, \delta \mu_n^k]^T \), \( e_k^k = [e_1^k, e_2^k, \ldots, e_n^k]^T \).

Given \( U = [\Delta \mu_1^1, \Delta \mu_2^2, \ldots, \Delta \mu_{N_p}^{N_p}]_{N \times N} \), \( E = [e_1^1, e_2^1, \ldots, e_n^1]^T \), we obtain
Then the perspective transform parameters can be obtained as:

$$A_h = UE^T(EE^T)^{-1}$$

(6)

Then the perspective transform parameters can be obtained as:

$$\Delta \mu(t+\tau) = A_h[I(x, t_h) - I(f(x; \mu(t)), t + \tau)]$$

(7)

Jacobian matrix $A_h$ here can only guarantee the convergence with in tiny scope. In order to ensure the stability for fast moving target, a multi-scale training tactics is proposed. We train $A_{h[i]}$, $i \in [1, m]$ corresponding different ‘disturbance’ in ascending order of size. Thus, geometric distortion caused by motion is divided into $m$ layers according to the amplitude of motion, each layer corresponds to one Jacobian matrix. Low-layer matrix is used for searching in a larger scope, and high-layer matrix for searching in a small scope. During online tracking stage, in the same order, it starts iterative searching from the lowest layer. The priori error Jacobian matrix obtained from the training phase can guarantee the convergence in a large scope, but not accurate. Based on this result, it continue to use Jacobian matrix of the upper layer for accurate search. Repeat the process until it converges at the top layer.

The multi-scale hierarchical idea can greatly increase the convergent range to deal with high speed moving target. Jacobian matrices corresponding prior knowledge of different motion scope during training process guarantee the avoidance of local extremum, which means tracking failure. However, this method is essentially a global feature tracking method, it usually falls down to effectively describe the appearance when the target is partially occluded. In the next section, we try to improve the stability of L-K algorithm by using local information.

**Pyramid L-K Tracking Based on Local Feature**

Given two consecutive frames $I(x, y)$ and $J(x, y)$, the target tracking problem can be transformed into optimization problem of residual $e(d)$ the optimization problem according L-K optical flow.

$$\hat{d} = \arg \min_{\hat{d}} e(d) = \arg \min_{\hat{d}} \left( \sum_{i=1}^{u_1+w_1} \sum_{y=u_2-w_2}^{y=u_2+w_2} \left( I(x, y) - J(x+d_x, y+d_y) \right)^2 \right)$$

(8)

where $d = (d_x, d_y)$ is the translation vector. In actual processing, iterative search usually achieves more accurate solution, and it is widely applied.

L-K optical flow algorithm performs well both in complexity and stability, but it only works when motion between adjacent frames is small enough. Therefore, multi-resolution L-K tracking based on pyramid is adopted to deal with high-speed target.

The basic idea of this method is that, Gauss Pyramid of consecutive frames are built at first, the higher layer in pyramid is the down-sampling of the lower layer through Gauss smoothing. When the image is down-sampled to a certain layer, the motion between adjacent frames will become small enough to meet the constraints of iterative L-K method for local feature tracking. Tracking calculation starts from the top of the Pyramid, and gradually extended from the top down a layer. The tracking results achieved on the upper layer are projected as the initiation of the next layer. This process repeats until the motion in the original image is estimated. The whole process of the algorithm is shown in Fig.2.
Figure 2. Schematic diagram of Pyramid L-K tracking

The detailed steps of the algorithm are listed as following:

1. Construct the Gauss Pyramid of consecutive frames. Assuming the size of original video image $I$ is $w \times h$, and regarding it as the 0-th layer of Pyramid. Down-sample it through Gauss smoothing to generate first, second,..., $(n-1)$-th Pyramid image. For the $L$-th $(0 < L \leq n-1)$ layer, the gray level can be calculated as:

$$\begin{align*}
I^L(x, y) = G_{\text{kernel}} \otimes 
\begin{bmatrix}
I^{L-1}(-2, -2) & I^{L-1}(-1, -2) & I^{L-1}(0, -2) & I^{L-1}(1, -2) & I^{L-1}(2, -2) \\
I^{L-1}(-2, -1) & I^{L-1}(-1, -1) & I^{L-1}(0, -1) & I^{L-1}(1, -1) & I^{L-1}(2, -1) \\
I^{L-1}(-2, 0) & I^{L-1}(-1, 0) & I^{L-1}(0, 0) & I^{L-1}(1, 0) & I^{L-1}(2, 0) \\
I^{L-1}(-2, 1) & I^{L-1}(-1, 1) & I^{L-1}(0, 1) & I^{L-1}(1, 1) & I^{L-1}(2, 1) \\
I^{L-1}(-2, 2) & I^{L-1}(-1, 2) & I^{L-1}(0, 2) & I^{L-1}(1, 2) & I^{L-1}(2, 2)
\end{bmatrix}
\end{align*}$$

where $I^{L-1}(0, 0) = I^{L-1}(2x, 2y)$.

2. Run iterative L-K tracking according to the order from high to low in Pyramid, and reconstruct the tracking results on the upper layer as the initial value of the next layer. For the same layer of adjacent frame $I^L(x, y)$ and $J^L(x, y)$, according Eq.(8), the tracking residual is calculated as follows:

$$
e^L(d^L) = \sum_{x = n_x}^{n_x + w_x} \sum_{y = n_y}^{n_y + w_y} \left( I^L(x, y) - J^L(x + g^L_x + d_x, y + g^L_y + d_y) \right)^2$$

where $g^L = (g^L_x, g^L_y)$ is the initial estimation of translation of the $L$-th layer, which can be obtained by the (L+1)-th layer.

3. Recursive calculation until the 0-th layer, finally we can get the tracking results of the translation vector:

$$d = g^0 + d^0 = \sum_{L=0}^{n-1} 2^L d^L$$

Assume that the maximum motion the iterative L-K algorithm can effectively track is $d_{\text{max}}$ in single layer, the maximum motion for Pyramid containing $m$ layers is

$$d_{\text{final}} = (2^m - 1) d_{\text{max}}$$
Experiments and Results

Our algorithm was implemented in visual studio 2016, which ran on an Intel i7-7700 3.5 GHz CPU with 8GB RAM without any optimizing. All the testing sequences have the same resolution of 640×480 pixels, and the targets include almost all states in actual applications, just like scale variations, partial occlusion, lightness change, rotation, affine deformation, etc.

Convergence Analysis. For multi-scale priori L-K tracking, we set the number of iterations per motion scale \( n_{itr} = 5 \). For the selected parameters, the convergence of the proposed algorithm is analyzed by comparing with the IC tracker.

For a given template image in Fig.3(a), the ground-truth of perspective transformation matrices between the input images which shown in Fig.3(b) and Fig.3(c) and the template image are:

\[
F_1 = \begin{bmatrix} 0.9424 & 0.1453 & -10.9864 \\ -0.1044 & 1.0211 & 19.0498 \\ -0.0001 & 0.0002 & 1 \end{bmatrix} \quad F_2 = \begin{bmatrix} 0.9403 & 0.2379 & -33.1093 \\ -0.0923 & 1.0483 & 13.9381 \\ -0.0001 & 0.0006 & 1 \end{bmatrix}
\]

From transformation matrix calibrated by perspective \( F_1 \), it can be seen that the displacement and deformation of the target area relative to the template image are relatively small in the input image 1, so both methods converge well. The IC algorithm achieves exact matching after 26 iterations, and our algorithm (MPIC) converges after 25 iterations. The tracking results and the iterative process are shown in Fig.3(b) and (d) respectively. In Fig.3(b), the tracking results boxes almost completely coincide.

For the input image 2 shown in Fig.3(c), the displacement and deformation of the target area relative to the template image is relatively large. Our algorithm can still converge to the exact position after 25 iterations, while the IC fails the tracking due to the local minimum. The tracking results and the iterative process are shown in Fig.3(c) and (e) respectively.

Local Feature and Its Validity. As for adjacent frames in testing video sequence shown in Fig.4(a), we detect 200 FAST-9 corner points, and use the iterative L-K feature tracker with 5 layers Pyramid. In order to improve the tracking speed, the translational motion model is used to constraint points tracking quality instead of affine motion model. It is obvious to find that our algorithm successfully tracks 117 feature points under the the maximum 70 pixel offset between the feature
points of two images. As shown in Fig.4 (b), our algorithm matched 87 feature points under large motion and motion blur.

![Pyramid L-K tracking results. (a) tracking results under movement, (b) tracking results under motion blur.](image1)

For the iterative Pyramid L-K tracking results, we take use of RANSAC method to eliminate the error tracking points, and estimate the homography transformation relation between adjacent frames, then we realize tracking for the original target template.

![Tracking results of a city model.](image2)

Robustness. Test sequence shown in Fig.5 includes the airport, port, bridges and urban buildings in a sandbox, which simulates real environment disturbance, including the target rotation, partial occlusion, scale variation. The tracking results are marked in a rectangular box. It is obvious that, benefits from local features, even when the target suffers from partially occlusion, our algorithm can still maintain stable tracking. The main reason for tracking failure is that most areas of the target are out of view. Our algorithm finally achieved 96.5% success rate on this test video.

Conclusion

In this paper, we propose an improved tracking method based on multi-scale motion prior information. The multi-scale priori error Jacobi matrix is calculated offline by artificially constructed multi-scale disturbance. The target searching strategy based on multi-scale stratification can greatly increase the convergence range of the algorithm. The pyramid iterative L-K tracking method based on local feature is employed to improve the stability of global feature tracking in the case of local occlusion. Experimental results demonstrate the validity of our method.

Acknowledgement

This work has been supported by the National Natural Science Foundation of China (Grant: 61403130 and 61402152), the Foundation of Key Laboratory of control engineering of Henan Province, Henan Polytechnic University (Grant: KG2014-17), the Doctor Foundation of Henan Polytechnic University (Grant: B2012-060).

Reference


