

A New, Self-Adaptive, KLT-based Algorithm for Visual Tracking

Cai Xuejun

*School of Computer Engineering, Shenzhen Polytechnic, Shenzhen, Guangdong
518055
caixuejun@szpt.edu.cn*

Abstract

Based on the analysis of the shortcomings of the current KLT (Kanade-Lucas-Tomasi) algorithm, a new, self-adaptive tracking algorithm is devised. Through introducing a kind of filtering mechanism, interference from surrounding noise and light on the tracked target is reduced. In order to reduce the error in tracking, a method based on forward-backward error approximation is utilized. Since such approximation reduces the number of visual feature points on the target, when the target changes in its shape tracking failure may easily result. In order to prevent this possible failure, a mechanism that appends the feature points is introduced. Experimental comparisons show that the said algorithm is clearly more effective than other similar algorithms.

Keywords: *Visual Tracking, Self-Adaptive, forward-backward error approximation, KLT*

1. Introduction

Visual tracking refers to the examination, extraction, recognition and tracking of moving objects in a series of frames of images in order to obtain the parameters of their motions—such as position, speed, acceleration and trail of motion—for further analysis, and to accomplish an understanding of the moving objects' behavior for tasks of higher order examination. Visual tracking can be widely applied in such practical fields as agriculture and defense in the following four aspects: visual surveillance, image compression, image stitching and 3-dimensional reconstruction. It has hence been one of the core issues in machine vision, and an area of keen interest to researchers both at home and abroad.

There exist a lot of visual tracking algorithms. Some of the more popular ones include Mean-Shift, CamShift (Continuously Adaptive Mean-SHIFT) and KLT. Mean-Shift is based on grayscale histogram and maximum similarity measure. It is highly responsive yet unable to resolve the problems of template update and target scaling, a trait upon which Document 10 endeavors to improve by introducing the newer, self-adaptive Mean-Shift algorithm with template update. It utilizes the Mean-Shift algorithm to process and track the target, and the least squares model of the results from SIFT matching of feature points to solve for coefficients and update the target. Experiments have shown that this algorithm manifests significant robustness when processing a target going through considerable variations in its dimensions. Because of the application of Mean-Shift, however, when background color is similar to that of the target, or when there are objects of similar color in the vicinity of the target, this algorithm will automatically include them, which leads to the expansion of the tracking window, sometimes to the extent of covering the entire frame, severely compromising the tracking precision and efficacy. This algorithm is therefore more suited to environments with ideal backgrounds. When dealing with complex environments it is hard to be self-adaptive. Furthermore, it involves large amount of calculation and is poor in its real-time performance, on account of its use of the SIFT (Scale Invariant Feature Transform) feature.

KLT tracking algorithm is the most popular sparse optical flow tracking method that is precise and highly reactive. A shortcoming of KLT is that, when it is applied to a small window and when the magnitude of motion is significant, feature points will move out of the window, resulting in tracking failure. Researchers have hence introduced the KLT algorithm based on pyramid, which can sufficiently overcome this difficulty. Because of the very morbid nature of tracking itself, of the constant variance in brightness of real environments, and of the fact that the assumption of spatial consistency is not necessarily always satisfied, however, tracking error seems never avoidable.

In order to solve the problems present in current tracking methods, this treatise proposes a new KLT-based, self-adaptive robust tracking algorithm. The said algorithm incorporates the mechanism of forward-backward error approximation that reduces tracking error and failure, adopts a filtering mechanism to increase the robustness of the system and lessen the impact of fluctuations in noise or surrounding light, and increases feature points to offset the negative impact brought about by forward-backward error approximation. Moreover, it utilizes a metric function that tests the stability of trackable points to further prevent tracking failure due to errors in tracking those points. Experiments show that this algorithm is far more advanced than CamShift and KLT in terms of robustness.

2. Mechanism of KLT Algorithm

KLT algorithm is based on the following three assumptions: (1) constant brightness, i.e. the brightness of tracked pixel points does not change; (2) continuous time or limited motion by the target between consecutive frames; (3) consistent space, i.e. within a small space w motion vector remains constant or tracked target maintains similar motion throughout.

KLT algorithm bases on the metric of the gray SSD (Sum of Squared Difference) of tracking window w . That is

$$e = \iint_w \left(J\left(X + \frac{d}{2}\right) - I\left(X - \frac{d}{2}\right) \right)^2 w(X) dX \quad (1)$$

where $X=[x, y]^T$ is expressed as the feature-point position, $d=[d_x, d_y]^T$ represents the displacement vector of the feature point, d_x and d_y mean the displacements in the directions of d_x and d_y , respectively, and $W(x)$ is a weighted function. If emphasis is placed on the grains in the central section, a Gaussian function can be set but usually at the value of 1. J and I are the gray values of pixels within the rectangular window w at times $t+1$ and t , respectively. Thus

$$\frac{\partial e}{\partial d} = 2 \iint_w \left(J\left(X + \frac{d}{2}\right) - I\left(X - \frac{d}{2}\right) \right) \left(\frac{\partial J\left(X + \frac{d}{2}\right)}{\partial d} - \frac{\partial I\left(X - \frac{d}{2}\right)}{\partial d} \right) w(X) dX \quad (2)$$

If displacement d is small, $J(X + d/2)$ and $I(X - d/2)$ can be expanded according to the first-order Taylor formulation and simplified as follows:

$$\frac{\partial e}{\partial d} \approx \iint_w \left(J(X) - I(X) + g^T d \right) g(X) w(X) dX \quad (3)$$

where $g = \left[\frac{\partial}{\partial x} \left(\frac{I+J}{2} \right), \frac{\partial}{\partial y} \left(\frac{I+J}{2} \right) \right]^T$

Further set

$$d = \arg \min_d e = \arg \min_d \iint_w (J(X + \frac{d}{2}) - I(X - \frac{d}{2}))^2 w(X) dX \quad (4)$$

In order to obtain the minimal value e , Let $\frac{\partial e}{\partial d} = 0$, then

$$\begin{aligned} \iint_w (I(X) - J(X)) g(X) w(X) dX &= \iint_w g^T(X) dg(X) w(X) dX \\ &= [\iint_w g(X) g^T(X) w(X) dX] d \end{aligned} \quad (5)$$

Let

$$A = \iint_w g(X) g^T(X) w(X) dX \quad (6)$$

$$b = (I(X) - J(X)) g(X) w(X) dX \quad (7)$$

Then

$$Ad=b \quad (8)$$

If matrix A is reversible, then

$$d=(dx, dy)=A^{-1}b \quad (9)$$

To ensure the matrix A is invertible, as a rule, corners are selected as feature points rather than image edges. Therefore, according to equation (9) we may achieve the displacement of the feature point.

3. Self-adaptive Tracking Mechanism

Though the current KLT has its advantages in speed and precision, it manifests several weaknesses summarized as follows:

(1) When displacement d is too large, KLT can lose track of the target and drift. This is because when equation (2) is to be expanded into its 1st-order Taylor series d needs to tend towards 0. Otherwise $Ad=b$ does not stand valid. It is common for d to be large for, say a 30 Hz camera, in reality. Therefore tracking failure results often.

(2) KLT requires constant brightness. When there is interference from noise margin of error is increased. Thus the requirement of equation (1) equaling 0 is not satisfied, affecting the tracking of target. Furthermore, inconstant brightness occurs very often in real life, which translates into poor robustness of the tracking system.

(3) Feature points may result in the kind of tracking error illustrated by Figure 1 due to blocking or change in shape in the target. In this example it can be seen that ① and ② are the feature points being tracked. While the system is still able to keep track of ①, ② is lost as a result of a change in the image.

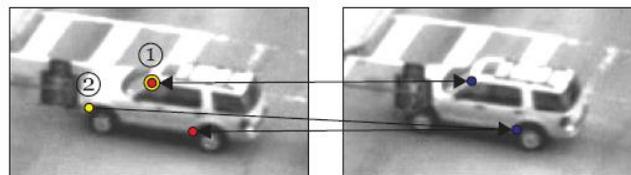


Figure 1. An Example of Tracking Failure

(4) It does not run any reliability test on the tracked points. Nor does it know whether tracking failure has not occurred with the feature points. Hence when it has, system may still continue to track.

In order to prevent tracking failure when displacement d is large, this treatise proposes the use of image pyramid, which is to first start tracking in the initial, larger space and then through downward image pyramid at its image pixel processing level to rectify the hypothesis on the initial velocity of motion. At the highest level of the pyramid optical flow is to be calculated. The estimation on the motion is then to be used as the starting point of the next level of the pyramid. Repeat this process until reaching the bottom level so as to realize tracking of objects under high speed.

In order to prevent noise or inconstancies in brightness from affecting the stability of the tracking system and to improve its robustness, a filtering mechanism is to be adopted. It involves selecting multiple feature points within the target, calculating their respective displacements, removing those that are too large or too small, and using their mean to reduce interference from noise or inconstant brightness.

When it comes to combating tracking failure arising from blocking or changes in shape, this treatise proposes the use of forward-backward error approximation. First calculate the displacements of the feature points from times t to $t+1$, during which image I changes into image J , using the KLT algorithm. Then, using the feature points in J as template, back-calculate the positions of corresponding feature points. As shown by Figure 1 where feature points ① and ② are taken as the tracked points, obtain the feature points using KLT in image J to perform the backward KLT. Then, using the feature points in J as template and the same algorithm, calculate the positions of the feature points in the previous frame I to perform the forward KLT. As can be known by Figure 1, with respect to feature point ① its position remains unchanged after backward KLT, and hence its distance 0; with respect to feature point ②, however, after backward KLT its position changes, the feature point found is no longer ② and its distance is hence not 0. It is no longer trackable. As a measure to improve the robustness of the system, a threshold value is to be set. Anything less than the threshold is to be recognized as a tracked feature point. Otherwise it is to be discarded. In so doing the negative impact from blocking or changes in shape can be removed. As tracking proceeds, however, and the target's mode of shape change varies, feature points can decrease progressively under this mechanism, resulting in tracking failure. Therefore a feature point appending mechanism is further incorporated, in which a point still is to be tracked if it is located within or in close proximity to a valid feature point, even if the forward-backward error is greater than the set threshold. This mechanism can further ensure the efficacy of the tracking system.

To ensure the reliability of tracked points, a assessment scheme based on similarity measure is introduced. It operates according to the following equation:

$$\rho = \frac{\text{cov}(x_i, x_j)}{\sqrt{D(x_i)D(x_j)}} \quad (10)$$

where x_i and x_j denote the areas covered by feature points in frames I and J , respectively; $\text{cov}(x_i, x_j)$ the covariance and D the variance. Only points whose reliability is greater than a certain threshold value can be tracked, which further enhances the tracking system's stability and reliability.

4. The KLT-based self-adaptive Algorithm

Based on the above-mentioned tracking mechanisms, this treatise proposes the following algorithm:

Step 1: Initialization

(1) Determine such parameters as size of tracking window w , distance threshold between feature points, tracking error, maximum number of tracked corners, proportion of ascertained interest area, minimum number of tracked points and number of pyramid levels.

(2) Input the first frame of image in the series and manually select the target to be tracked. Extract KLT corner points to obtain the size of the area of interest.

Step 2: Tracking

(1) Select corner feature points. If number of corners is greater than maximum number of tracked points, calculate the Eigenvalue of matrix A , based on which the selection is made.

(2) If number of feature points is less than minimum number of tracked points, skip to Step 5.

(3) With respect to the interest area, establish a multi-resolution Gaussian pyramid image based on the number of pyramid levels obtained in initialization.

(4) Use approximation acquired at an upper level as the starting point of its subsequent lower level. By way of iteration complete the KLT tracking algorithm.

Step 3: Feature point selection

(1) Using the filtering mechanism proposed in the previous chapter, exclude feature points with displacements that are too large or too small.

(2) Using the method of forward-backward error approximation mentioned in the previous chapter, calculate the error and remove points that are erroneously tracked.

(3) Calculate reliability of feature points according to equation (10) and carry out feature point selection.

(4) If number of remaining feature points is greater than the initial threshold, use those points to calculate a mean to be used as the displacement of the moving target and skip to Step 4. Otherwise skip to Step 5.

Step 4: Feature point preservation

If the feature points selected in part (2) of Step 3 are within or in close proximity to valid feature points, they are still to be used as the tracked in the next frame and return to Step 2, regardless of whether the forward-backward error is not greater than the set threshold.

Step 5: Exit program.

5. Experimental Results and Analysis

In order to assess the efficacy of the algorithm proposed in this treatise, many evaluative methods have been used in the experiments conducted.

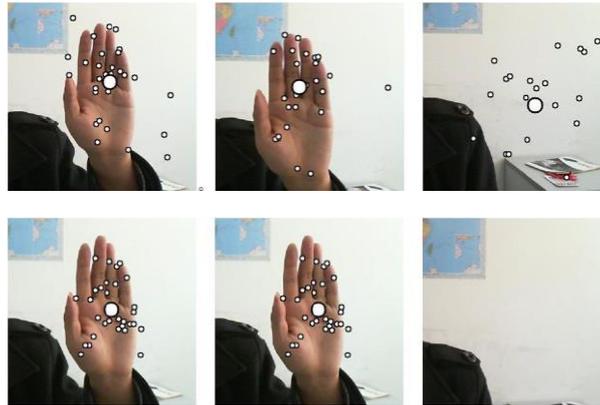


Figure 2. Comparison between Two Sets of Real-Time Online Results

Figure 2 shows a segment of real-time online tracking based on VS 2008 platform, in which the first and the second rows are the processing results obtained through regular KLT algorithm and through the algorithm proposed herein, respectively. The white dots denote the feature points being tracked. This comparison shows that, with the lack of forward-backward error approximation and reliability test on feature points, regular KLT still maintains tracking of even false targets when the actual target is absent, hence fails in the tracking task.

Figure 3 shows a segment of video processed on Matlab with slight variation in brightness and a touch of added noise, in which the first, second and the third columns are the 267th, 269th and 270th frames of the series, respectively; and the first row is the results from regular KLT and the second, the new algorithm proposed by this treatise. It can be seen that the latter is more effective than regular KLT, with its filtering mechanism that reduces interference from noise and surrounding light.

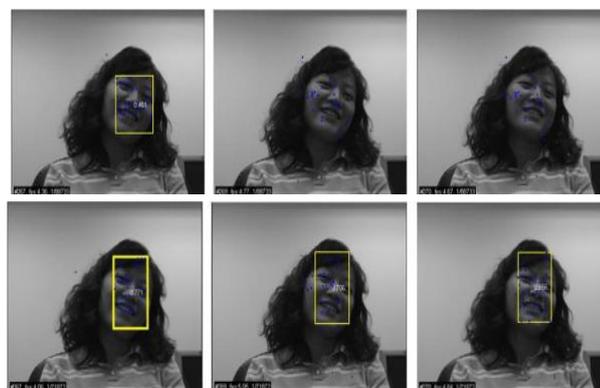


Figure 3. Comparison Offline KLT and the New Proposed Algorithm

In order to see with greater clarity this new tracking algorithm's advantage, a comparison is made with the traditional CamShift algorithm. As shown by Figure 4, in which the first row is the results obtained from CamShift, and the second row from the new algorithm, the new algorithm has produced more precise results than has the traditional CamShift. This is on account of the similarity between the color histograms of the target face and of the hand. When the hand is in close proximity to the face, the tracking area of CamShift expands almost to the entirety of the frame, thus resulting in lower precision.



Figure 4. Comparison between CamShift and the New Algorithm

6. Conclusion

The self-adaptive tracking algorithm proposed in this treatise is able to overcome many of the shortcomings of the regular KLT. Experimental results show that this new algorithm is more advanced than KLT and CamShift in terms of precision, reliability and robustness, owing to the following mechanisms: (1) filtering mechanism that reduces the impact of noise and surrounding light on KLT algorithm; (2) forward-backward error approximation and reliability test that precludes continued erroneous tracking; (3) feature point appending mechanism that offsets loss of feature points while target undergoes changes in shape.

A weakness of the proposed algorithm is that when blocking occurs to the target, tracking failure will ensue. This is on account of the lack of an online detection mechanism. It is hence important to carry out researches on detection and tracking algorithms based on online learning at next stage.

Acknowledgements

The work is supported by the Guangdong Provincial Natural Science Foundation (Grant No.S2013010015471, S2013010016606), and the Shenzhen Basic Research Project (Grant No.JCYJ20120617140737337, Grant No.JCYJ20130331151803073).

References

- [1] H. Zhi-Qian and H. Chong-Zhao, "A Survey of Visual Tracking", ACTA AUTOMATICA SINICA, vol. 32, no. 4, (2006), pp. 603-617.
- [2] I. Haritaoglu, D. Harwood and L. S. Davis, "W4: Real-time surveillance of people and their activities", IEEE Transactions on Pattern Analysis and Machine, vol. 22, no. 8, (2000), pp. 809-830.
- [3] Z. Zhong-jie, J. Gang-yi and Y. Mei, "New Algorithm for Extracting and Tracking Moving Object in Object-Based Video Coding", ACTA ELECTRONICA SINICA, vol. 31, no. 9, (2003), pp. 1426-1428.
- [4] H. She-jia, G. Xi-wang and C. Zong-hai, "Based on Corner Feature KLT Track Panoramic Mosaic Algorithm", Journal of System Simulation, vol. 19, no. 8, (2007), pp. 1742-1745.
- [5] J. Y. Zheng and A. Murata, "Acquiring a complete 3D model from specular motion under the illumination of circular-shaped light sources", IEEE Transactions on Pattern Analysis and Machine, vol. 22, no. 8, (2000), pp. 913-920.
- [6] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, (2002), pp. 603-619.
- [7] G. R. Bradski, "Computer Vision Face Tracking For Use in a Perceptual Use interface", Intel Technology Journal, vol. 2, no. 2, (1998), pp. 1-15.
- [8] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision", International Joint Conference on Artificial Intelligence, (1981), pp. 674-679.
- [9] C. Tomasi and T. Kanade, "Detection and tracking of point features", Tech. Rep. CMU-CS-91-132, Carnegie Mellon University, (1991), pp. 1-22.
- [10] L. Yong-yong, T. Yi-hua and T. Jin-wen, "Template Updating Based Adaptive Tracking Algorithm Using Mean-shift", Computer Science, vol. 38, no. 3, (2011), pp. 271-274.

- [11] D. G. Lowe, "Distinctive image features from scale-invariant key-points", International Journal of Computer Vision, vol. 60, no. 2, (2004), pp. 91-110.
- [12] G. Bradski and A. Kaebler, "Learning openCV computer vision with the OpenCV library", O'Reilly (2008), pp. 323-334.
- [13] Z. Kalal, K. Mikolajczyk and J. Matas, "Forward-backward error: automatic detection of tracking failures", International Conference on Pattern Recognition, (2010), pp. 2756-2759.
- [14] J. Shi and C. Tomasi, "Good features to track", IEEE Conference on Computer Vision and Pattern Recognition, (1994), pp. 593-600.

Author

CAI Xuejun, Professor Cai has a specialty in computer algorithm, information security, and internetworking.