Research Article

Least Trimmed Squares Approach to Lucas-Kanade Algorithm in Object Tracking Problems

Yih-Lon Lin

Department of Information Engineering, I-Shou University, Kaohsiung 84001, Taiwan

Correspondence should be addressed to Yih-Lon Lin; yllin@isu.edu.tw

Received 11 April 2013; Accepted 19 June 2013

Academic Editor: Ker-Wei Yu

Copyright © 2013 Yih-Lon Lin. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The object tracking problem is an important research topic in computer vision. For real applications such as vehicle tracking and face tracking, there are many efficient and real-time algorithms. In this study, we will focus on the Lucas-Kanade (LK) algorithm for object tracking. Although this method is time consuming, it is effective in tracking accuracy and environment adaptation. In the standard LK method, the sum of squared errors is used as the cost function, while least trimmed squares is adopted as the cost function in this study. The resulting estimator is robust against outliers caused by noises and occlusions in the tracking process. Simulations are provided to show that the proposed algorithm outperforms the standard LK method in the sense that it is robust against the outliers in the object tracking problems.

1. Introduction

The Lucas-Kanade (LK) algorithm was originally proposed by Lucas and Kanade in 1981 [1], which makes use of the spatial intensity gradient of the images to find a good match using a type of the Newton-Raphson iteration. The goal of the standard LK algorithm is to minimize the sum of squared errors (SSE) function between the template and the warped image region by adjusting the warping parameters.

Based on the theory of robust statistics, various types of cost functions were adopted in many practical applications such as regression. One of the main approaches to robust regression involves M -estimation [2–6]. The M -estimator is found to be robust to outliers in the response variable, but turned out not to be resistant to outliers in the explanatory variables, called leverage points. In fact, when there are outliers in the explanatory variables, the method has no advantage over least squares. In the 1980s, several alternatives to M -estimation were proposed as attempts to overcome the lack of such resistance. Least trimmed (sum of) squares (LTS) is a viable alternative [7–10].

There are several robust LK algorithms proposed in the following literatures. The M -estimators were adopted to LK models for tracking problems in [11–15]. The basic idea is to replace the cost function SSE by Huber or Hampel function

so that the effects of outliers may be degraded. In [11], the authors generalized the LK approach to histogram-based tracking algorithm. It establishes a closer link between template matching and histogram-based tracking methods. Schreiber [12] proposed a novel variant of the LK algorithm for tracking bilaterally symmetric planar objects from a moving platform. This algorithm was capable of coping with any warping transformation and can be generalized for the case of objects possessing higher symmetry. According to analysis of data distributions, Senst et al. [13] proposed a robust local optical flow approach based on a modified Hampel estimator with robust characteristics. Fan et al. [14] proposed a robust template tracking with weighted active drift correction. The minimization of active drift correction is achieved by the inverse compositional algorithm, which consists of the tracking term and the drift correction term. In [15], a robust LK template matching algorithm was based on evidence which is accumulated over many frames. This algorithm described the drift correcting and used robust weights that are being updated from frame to frame. The previous papers emphasized robust inverse composition algorithm using a robust Huber or Hampel cost functions. However, there are many other methods for robust LK algorithm such as robust measurement of ocular torsion [16], visual tracking and learning [17], and graph-based transductive learning [18].

The LTS estimator is highly resistant to leverage points and is robust to outliers in the response. When we expect there to be some number of observations in the data that we wish to put no weight in the modeling, LTS estimator is usually a good choice. The percentage of the data that we wish to put no weight is termed the trimming percentage or trimming parameter, and this parameter is usually prespecified for the data at hand. In this study, the main idea is to utilize the resistant property of the LTS estimators in the LK algorithm (LTS-LK) for object tracking problems when there are outliers caused by, for instance, noises and occlusions. The updating rules for the warping parameters using LTS-LK approach will be derived. Three simulations are provided in this paper, two for book-object tracking and one for face-object tracking. The simulation results show that the proposed method can effectively track the object when salt-and-pepper noises corruption or occlusion appears.

The rest of this paper is organized as follows. In Section 2, we outline the standard LK algorithm. Section 3 presents the detail of the proposed LTS-LK method. Some experimental results are provided in Section 4 to verify the performances of the proposed method. Finally, a conclusion is made in Section 5.

2. Lucas-Kanade Algorithm

Template tracking of an object in a video sequence is performed by extracting a template in the first frame and then finding the region which matches the template as close as possible in the surplus frames. The goal of LK method is to align a template image $T(\mathbf{x})$ to an input image $I(\mathbf{x})$, where $\mathbf{x} = [x \ y]^T$ is a column vector representing the pixel coordinates. Let $\mathbf{W}(\mathbf{x}; \mathbf{p})$ denote the parameterized set of allowed warps, where $\mathbf{p} = [p_1 \ \dots \ p_6]^T$ is a vector of warping parameters. Here, we consider the following set of affine warps:

$$\mathbf{W}(\mathbf{x}; \mathbf{p}) = \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{bmatrix} = \begin{bmatrix} 1 + p_1 & p_3 \\ p_2 & 1 + p_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} p_5 \\ p_6 \end{bmatrix}. \quad (1)$$

The best match to the template in the new frame is found by minimizing the following SSE function:

$$E = \sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - T(\mathbf{x})]^2, \quad (2)$$

where $\mathbf{W}(\mathbf{x}; \mathbf{p})$ is defined in (1) and the sum is performed over all of the pixels $\mathbf{x}$ in the template image $T(\mathbf{x})$. Based on the SSE function in (2), the goal of the LK algorithm is therefore to find

$$\mathbf{p}_{n+1} = \arg \min_{\mathbf{p}} \sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x}; \mathbf{p}_n)) - T(\mathbf{x})]^2, \quad \text{for } n \geq 1, \quad (3)$$

where n is the frame number. Suppose that a current estimate of $\mathbf{p}$ is known and we wish to compute an appropriate increment $\Delta \mathbf{p}$. The minimization problem in (3) can then be converted to

$$\arg \min_{\Delta \mathbf{p}} \sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x}; \mathbf{p} + \Delta \mathbf{p})) - T(\mathbf{x})]^2. \quad (4)$$

Performing a first-order Taylor series expansion on $I(\mathbf{W}(\mathbf{x}; \mathbf{p} + \Delta \mathbf{p}))$ at $\Delta \mathbf{p} = 0$ in (4) yields

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x}; \mathbf{p}) + \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p}) - T(\mathbf{x}) \right]^2, \quad (5)$$

where $\nabla I = [\partial I / \partial x \ \partial I / \partial y]^T$ is the gradient of the image I evaluated at $\mathbf{W}(\mathbf{x}; \mathbf{p})$. The term $\partial \mathbf{W} / \partial \mathbf{p}$ is the Jacobian of the warp given by

$$\begin{aligned} \frac{\partial \mathbf{W}}{\partial \mathbf{p}} &= \begin{bmatrix} \frac{\partial \mathbf{W}_1}{\partial p_1} & \frac{\partial \mathbf{W}_1}{\partial p_2} & \dots & \frac{\partial \mathbf{W}_1}{\partial p_6} \\ \frac{\partial \mathbf{W}_2}{\partial p_1} & \frac{\partial \mathbf{W}_2}{\partial p_2} & \dots & \frac{\partial \mathbf{W}_2}{\partial p_6} \end{bmatrix} \\ &= \frac{\partial}{\partial \mathbf{p}} \begin{bmatrix} x + xp_1 + yp_3 + p_5 \\ xp_2 + y + yp_4 + p_6 \end{bmatrix} \\ &= \begin{bmatrix} x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1 \end{bmatrix}. \end{aligned} \quad (6)$$

An approximate solution to (4) can be obtained by taking the partial derivatives of (5) with respect to $\Delta \mathbf{p}$ and setting them to zero. It is easy to derive that this approximate solution is given by

$$\Delta \mathbf{p} = H^{-1} \sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T [I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - T(\mathbf{x})], \quad (7)$$

where H is Gauss-Newton approximation to the Hessian matrix

$$H = \sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]. \quad (8)$$

Finally, the parameters are updated as

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta \mathbf{p}, \quad (9)$$

and the process is iterated until the estimate of parameters $\mathbf{p}$ converge or the pre-specified stopping criterion is met.

3. LK Algorithm Using LTS Approach

Let $X \subseteq \mathfrak{R}^n$ and $Y \subseteq \mathfrak{R}^p$. Suppose we are given the training set

$$S := \{(x_q, d_q)\}_{q=1}^l \subseteq X \times Y. \quad (10)$$

In the following, we will use the subscript q to denote the q th observation. For instance, x_{qi} denotes the i th component of the q th input $x_q \in \mathfrak{R}^n$, $q \in \underline{l}$, $i \in \underline{n}$. The residual (or error) e_{qk} at the k th component of the difference between the desired output d_{qk} and the predicted output y_{qk} due to the q th observation is defined by

$$e_{qk} := d_{qk} - y_{qk}, \quad q \in \underline{l}, \quad k \in \underline{p}. \quad (11)$$

The LTS approach is to choose parameters that minimize the total sum of trimmed squared errors given by

$$\begin{aligned}\Psi_{\text{total}} &= \frac{1}{2} \sum_{k=1}^p \sum_{q=1}^{l^*} e_{(q)k}^2 \\ &= \frac{1}{2} \sum_{k=1}^p \sum_{q=1}^l a(R(e_{qk}^2)) e_{qk}^2,\end{aligned}\quad (12)$$

where $l^* \leq l$ and the penalizing weight $a(j)$ is defined by

$$a(j) := \begin{cases} 1, & 1 \leq j \leq l^*, \\ 0, & l^* < j \leq l. \end{cases} \quad (13)$$

Here, $R(e_{qk}^2)$ denotes the rank of the residual e_{qk}^2 , $q \in \underline{l}$, and $e_{(1)k}^2 \leq \dots \leq e_{(l)k}^2$ are the ordered values of $e_{1k}^2, \dots, e_{lk}^2$. A popular choice of l^* is

$$l^* = \left\lfloor \frac{l}{2} \right\rfloor + \left\lfloor \frac{(n+1)}{2} \right\rfloor, \quad (14)$$

where $[b]$ indicates the largest integer less than or equal to b . In the following, the term $(l - l^*)/l$ is referred to as the trimming percentage.

As given in the previous section, the values of $\Delta \mathbf{p}$ in (7) are computed from the inverse compositional algorithm. To adapt the notation for LTS model, we regard the template image $T(\mathbf{x})$ and subpixel location $\mathbf{W}(\mathbf{x}; \mathbf{p})$ as one-dimensional vectors d_i and z_i , $1 \leq i \leq l$, respectively, where l is the number of pixels in the template image. Now the training set becomes

$$S := \{(z_i, d_i)\}_{i=1}^l \subseteq \mathfrak{R}^2 \times \mathfrak{R}. \quad (15)$$

The predictive function, a two-dimensional image, is given by

$$f(\mathbf{x}) := I(\mathbf{W}(\mathbf{x}; \mathbf{p})). \quad (16)$$

It can be regarded as a one-dimensional vector $f(z_i)$. The residuals are given by

$$\rho_i := d_i - f(z_i), \quad i \in \underline{l}. \quad (17)$$

One of the commonly used cost function is the l_2 -norm of the residuals $\rho := [\rho_1 \dots \rho_l]^T \in \mathfrak{R}^l$. This is the cost function used in (4). To address the object tracking problems when there are outliers caused by, for instance, noises and occlusions, we will use sum of trimmed squared errors instead of the l_2 -norm. Consequently, the goal of the LTS estimator is to find the parameters $\Delta \mathbf{p}$ of LK algorithm that minimize the cost function defined by

$$\begin{aligned}E &= \sum_{\mathbf{x}} a(R([I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - T(\mathbf{x})]^2)) \\ &\quad \times \left[I\left(\mathbf{W}(\mathbf{x}; \mathbf{p}) + \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p}\right) - T(\mathbf{x}) \right]^2.\end{aligned}\quad (18)$$

FIGURE 1: Flowchart of LTS-LK algorithm.

It is not difficult to derive that the variable $\Delta \mathbf{p}$ can be calculated as

$$\begin{aligned}\Delta \mathbf{p} &= H_T^{-1} \sum_{\mathbf{x}} a(R([I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - T(\mathbf{x})]^2)) \\ &\quad \times \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T [I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - T(\mathbf{x})],\end{aligned}\quad (19)$$

where the robust Hessian matrix is

$$\begin{aligned}H_T &= \sum_{\mathbf{x}} a(R([I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - T(\mathbf{x})]^2)) \\ &\quad \times \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right].\end{aligned}\quad (20)$$

The flowchart of the proposed LTS-LK algorithm is shown in Figure 1. Note that the linearized formulation (18) may be regarded as a linear weighted least squares regression problem. In practice, the inverse of Hessian matrix may pose some numerical difficulties in six-dimensional space. For higher dimension, recursive weighted least squares algorithm can be applied to solve this optimization problem. Here, we adopt direct computation of the inverse.

4. Experimental Simulation

In this section, we compare the performances of standard LK and the LTS-LK algorithms for object tracking. Emphasis is put particularly on the robustness against outliers. Simulation

FIGURE 2: Results of “book-tracking” sequence. Salt-and-pepper noises are added after Frame 100.

FIGURE 3: Results of “book-tracking” sequence. Occlusion is added after Frame 25.

programs are implemented using Borland C++ Builder 6.0 running on Microsoft Windows 7, Intel Core 2 Quad CPU, and 4 GB RAM platform.

To demonstrate the robustness of the proposed algorithm, we present three object-tracking experiments, consisting of two “book-tracking” and one “face-tracking” video sequences. In Example 1, the frames after the 100th of the “book” sequence are corrupted by salt-and-pepper noises in which a corrupted pixel has the intensity of 0 or 255. In Example 2, we add another type of corruption, a severe occlusion of about 1/3 template area at the 35th frame. Example 3 shows the same situation on face tracking with occlusion at the 240th frame. The trimming percentages used in our simulations are 5%, 20%, and 35% in the three examples, respectively. In the simulations, we set 20 as the maximal number of iterations and 0.0001 as the minimal change of Δp . The iteration stops when either of the two conditions is met.

Example 1. The “book” sequence is of size 320×240 containing 250 frames in which the object to be tracked is the book. In this experiment, the initial bounding box size is 60×80 pixels and contains about 6.25% of the frame size. Simulation results are shown in Figure 2. The first and second rows show the results of the standard LK and the proposed LTS-LK algorithms, respectively. To demonstrate the robustness of LTS-LK algorithm, we randomly corrupt the video sequence

with 20% noises, about 15,360 pixels, in the frames after the 100th frame. As observed, the standard LK algorithm fails to track the object in the 160th frame (the 3rd column), whereas our LTS-LK algorithm can effectively track the “book” object.

Example 2. In this example, the experiment deals with severe occlusions in another “book” sequence containing 60 frames, of size 320×240 . Initial bounding box size is 50×70 containing about 4.5% of its original size. The results are shown in Figure 3. As can be seen, the area of LK algorithm bounding box shrinks down to 50% of its original size at Frame 25 since occlusion appears, that is, because the standard LK algorithm is based on least-squares principle and is, therefore, sensitive to outliers. In the second row, we can see that the proposed LTS-LK algorithm on this sequence produces much better results when the object “book” suffers from severe occlusions. The mean squared error (MSE) of the tracked areas from LK algorithm and trimmed MSE from LTS-LK algorithm at the 35th frame are 8.95 and 1.76, respectively. As a consequence, LTS-LK algorithm demonstrates significant robustness under severe occlusion.

Example 3. The experiment deals with severe occlusions in “face-tracking” sequence containing 350 frames, of size 320×240 . The results are shown in Figure 4. In the LK algorithm, there appears numerical error on updating the variable Δp at the 240th frame. It means that the standard LK algorithm

FIGURE 4: Results of “face-tracking” sequence. Occlusion is added after Frame 100.

diverges. Therefore, the variable $\Delta \mathbf{p}$ needs not be updated after the 240th frame. As observed, the LK algorithm fails after the 240th frame, whereas our proposed algorithm can effectively track the “face” object. Again, the proposed LTS-LK algorithm is quite robust against severe occlusion.

From the illustrative examples above, when applying the LTS approach to object tracking problems, the proposed LTS-LK model is robust against outliers. This is achieved by choosing an appropriate trimming percentage. Unfortunately, there is no systematic way to select a proper trimming percentage for the general object tracking problem at hand. The choice is usually problem dependent.

5. Conclusion

The main issue of this paper is the robustness against outliers caused by noises and occlusions in the video object tracking problems. The proposed robust algorithm adopts the LTS scheme for the LK algorithm. Experimental results show that the proposed LTS-LK algorithm outperforms the standard LK. It is noted that SSE minimization criterion is used in standard approach to search the optimal parameters. However, it is known that the estimation of parameters based on least squares criterion is vulnerable to outliers. The main contribution of this paper is the use of LTS robust criteria, instead of the standard least squares criterion, for the search for the LK parameters, and this provides a robust object tracking application.

Acknowledgment

This research reported here was supported by the National Science Council, Taiwan, under Grant no. NSC 101-2221-E-214-074.

References

- [1] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 674–679, San Francisco, Calif, USA, August 1981.
- [2] J. W. Tukey, “A survey of sampling from contaminated distributions,” in *Contributions to Probability and Statistics*, I. Olkin, S. G. Ghurye, W. Hoeffding, W. G. Madow, and H. B. Mann, Eds., pp. 448–485, Stanford University Press, Palo Alto, Calif, USA, 1960.
- [3] P. J. Huber, “Robust estimation of a location parameter,” *Annals of Mathematical Statistics*, vol. 35, pp. 73–101, 1964.
- [4] P. J. Huber and E. M. Ronchetti, *Robust Statistics*, John Wiley & Sons, Hoboken, NJ, USA, 2nd edition, 2009.
- [5] F. Hampel, *Contributions to the theory of robust estimation [Ph.D. thesis]*, University of California, Berkeley, Calif, USA, 1968.
- [6] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach Based on Influence Functions*, John Wiley & Sons, New York, NY, USA, 2005.
- [7] P. J. Rousseeuw, “Multivariate estimation with high breakdown point,” in *Mathematical Statistics and Applications*, G. Pflug, I. Vincze, and W. Wertz, Eds., pp. 283–297, Reidel, Dordrecht, The Netherlands, 1985.
- [8] M. Tableman, “The influence functions for the least trimmed squares and the least trimmed absolute deviations estimators,” *Statistics & Probability Letters*, vol. 19, no. 4, pp. 329–337, 1994.
- [9] A. Giloni and M. Padberg, “Least trimmed squares regression, least median squares regression, and mathematical programming,” *Mathematical and Computer Modelling*, vol. 35, no. 9–10, pp. 1043–1060, 2002.
- [10] T. D. Nguyen and R. Welsch, “Outlier detection and least trimmed squares approximation using semi-definite programming,” *Computational Statistics & Data Analysis*, vol. 54, no. 12, pp. 3212–3226, 2010.
- [11] D. Schreiber, “Generalizing the Lucas-Kanade algorithm for histogram-based tracking,” *Pattern Recognition Letters*, vol. 29, no. 7, pp. 852–861, 2008.
- [12] D. Schreiber, “Incorporating symmetry into the Lucas-Kanade framework,” *Pattern Recognition Letters*, vol. 30, no. 7, pp. 690–698, 2009.
- [13] T. Senst, V. Eiselein, and T. Sikora, “Robust local optical flow for feature tracking,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 9, pp. 1377–1387, 2012.
- [14] B. Fan, Y. Du, L. Zhu, J. Sun, and Y. Tang, “A robust template tracking algorithm with weighted active drift correction,” *Pattern Recognition Letters*, vol. 32, no. 9, pp. 1317–1327, 2011.
- [15] D. Schreiber, “Robust template tracking with drift correction,” *Pattern Recognition Letters*, vol. 28, no. 12, pp. 1483–1491, 2007.

- [16] I. B. Lee, B. H. Choi, and K. S. Park, "Robust measurement of ocular torsion using iterative Lucas-Kanade," *Computer Methods and Programs in Biomedicine*, vol. 85, no. 3, pp. 238–246, 2007.
- [17] J. Li, Y. Wang, and Y. Wang, "Visual tracking and learning using speeded up robust features," *Pattern Recognition Letters*, vol. 33, no. 16, pp. 2094–2101, 2012.
- [18] Y. Zha, Y. Yang, and D. Bi, "Graph-based transductive learning for robust visual tracking," *Pattern Recognition*, vol. 43, no. 1, pp. 187–196, 2010.