Efficient Point Feature Tracking based on Self-aware Distance Transform

Min-Gyu Park
mpark@gist.ac.kr
Kuk-Jin Yoon
kjyoon@gist.ac.kr

Computer Vision Lab.
School of Information and Communications,
Gwangju Institute of Science and Technology (GIST),
Gwangju, Republic of Korea
http://cvl.gist.ac.kr

The tracking of point features is an essential problem in computer vision because the acquisiton of feature correspondence from successive frames is a front-end step in many problems. We divide feature tracking algorithm rougly into three categories, i.e., tracking-by-detection [2, 6], tracking-by-template matching [3, 5, 8], and tracking-by-Lucas-Kanade-Tomasi (KLT) [1, 4] tracker. In this paper, we focus on improving the second approach that inherits the intrinsic characteristics of the template matching problem. Previous studies focused mainly on increasing the speed of matching [3], improving the computational search efficiency [8], and developing robust similarity measures. However, the size of search region is still retained as a predefined parameter, although the size of a search region affects the performance of an algorithm significantly.

To tackle this problem, we propose a Self-aware Distance Transform (SDT) with an efficient feature-tracking method. The aim of the SDT is to estimate the optimal search region size based on the autocorrelation with a template in the initial frame. We use the spatial relationship of the cross-correlation coefficients relative to the best match as the function of a coefficient in the predicted position of a feature. After extracting a feature [7] and its corresponding template, we immediately perform autocorrelation of the image and the template; then generate a set of groups based on the distance to the best match as follow:

$$\begin{aligned} &F = \{F_1, F_2, ..., F_{M-1}, F_M\}, \\ &F_k = \{C(\mathbf{p}) | round(\sqrt{p_x^2 + p_y^2}) = k\} \text{ for } 1 \le k \le M, \end{aligned} \tag{1}$$

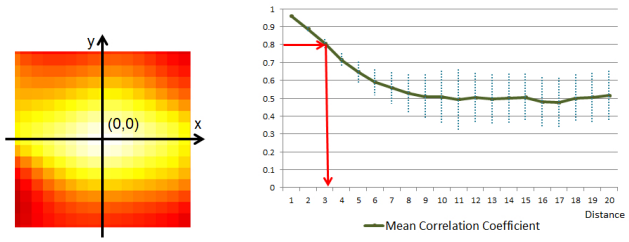
where F_k is a set of correlation coefficients with the same distance from the best match, $\mathbf{p} = [p_x \ p_y]^T$ is a relative position vector centered on the best match $(0,0)$, $C(\mathbf{p})$ is the autocorrelation coefficient at $\mathbf{p}$, and k ranges from 1 to the predefined constant M. This constant is tuned automatically during the last step, so the selection of this value is not a significant problem. Rather than using a continuous distance, we discretize the distance values for the group of pixels and this distance is computed using a Chamfer distance transform. Next, we compute the mean and variance of each group as follows:

$$\mu_k = \frac{1}{|F_k|} \sum_{C(\mathbf{p}) \in F_k} C(\mathbf{p}), \ \sigma_k^2 = \frac{1}{|F_k|} \sum_{C(\mathbf{p}) \in F_k} (C(\mathbf{p}) - \mu_k)^2, \tag{2}$$

where μ_k and σ_k indicate the mean and standard deviation of the autocorrelation coefficients at the distance k, while $|F_k|$ represents the cardinality of a set of correlation coefficients. These two statistics are the essence of SDT because they are used to the compute the optimal size of a search region. Figure 1 shows the autocorrelation result and the relationship between the mean and distance values. This relationship is used as a function of an NCC coefficient, which allows the size of a search region to be determined automatically at each prediction step. For example, Fig. 1 shows that the best match is probably within 3 pixels if the correlation value is 0.8. Finally, the SDT is defined as a function of a real valued vector (the position of a feature), which yields a positive integer value as follows:

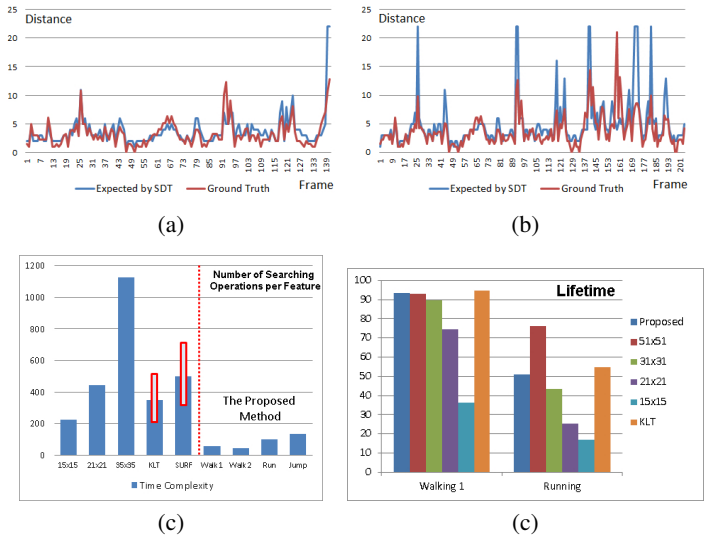
$$\begin{aligned} &SDT : \Re^2 \rightarrow N^+ \text{ s.t.} \\ &\hat{d}_{t+1} = arg\min_{1 \le k \le M} |\mu_k - C_{t+1}(\hat{\mathbf{x}}_{t+1})|, \end{aligned} \tag{3}$$

where $C_{t+1}(\hat{\mathbf{x}}_{t+1})$ indicates the NCC coefficient of a predicted position (we use a subscript to indicate that this computes the NCC between the template and a successive image at time $t+1$) while the expected distance is computed by minimizing the difference between the mean value and the NCC coefficients. Indeed, the SDT can be used for any other prediction models; we use the constant velocity model for the prediction. The expected distance contains uncertainty that is proportional to corresponding variance σ_k where k equals $\hat{d}_{t+1}$. To avoid unreliable estimation of expected distance, therefore, we restrict the range of the valid expected distance, $(0, d_{max}]$ is determined by thresholding larger variance values than a predefined threshold. For the experiment, we computed both the ground truth displacement and the predicted distance for the SDT evaluation. As shown in Fig. 2 (a–b), the SDT well approximates the actual displacement of features; thus, the size of a search region can be adaptively chosen. As a consequence, the time complexity of the proposed feature tracking method reduced significantly compared to other methods while maintaining a certain level of robustness against abrupt motion of features, as shown in Fig. 2 (c–d), respectively.



(a) (b)

Figure 1: Illustration of the SDT; (a) the result of autocorrelation and (b) the relationship between the mean values of the autocorrelation coefficients and distance. The dotted vertical line indicates the variance of the autocorrelation coefficients at the same distance.



(a) (b) (c) (c)

Figure 2: Comparison of the expected distances (blue lines) and the ground truth distances (red lines) for features in the walking 1 sequence (a–b), and the evaluation of the proposed method in terms of time complexity (c) and the lifetime of features (d) compared to other tracking methods.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2009-0065038).

[1] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework: Part 1. Technical Report CMU-RI-TR-02-16, Robotics Institute, Pittsburgh, PA, July 2002.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, pages 404–417, 2006.

[3] Yu-Wen Huang, Ching-Yeh Chen, Chen-Han Tsai, Chun-Fu Shen, and Liang-Gee Chen. Survey on block matching motion estimation algorithms and architectures with new results. *J. VLSI Signal Process. Syst.*, 42(3):297–320, March 2006.

[4] Myung Hwangbo, Jun-Sik Kim, and T. Kanade. Inertial-aided klt feature tracking for a moving camera. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1909 –1916, oct. 2009.

[5] J. P. Lewis. Fast normalized cross-correlation, 1995.

[6] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, November 2004. ISSN 0920-5691.

[7] Jianbo Shi and Carlo Tomasi. Good features to track. In *Computer Vision and Pattern Recognition (CVPR)*, Ithaca, NY, USA, 1993. Cornell University.

[8] Steven L Tanimoto. Template matching in pyramids. *Computer Graphics and Image Processing*, 16(4):356 – 369, 1981.