An Integrated Solution to Motion Tracking of Moving Vehicles Using Optical Flow

Dong-Min Woo, and An Nguyen Hong

***Abstract*—**Motion tracking is one of the most practical applications of computer vision in real life. In this paper, we highlight a new application for tracking motion and estimating the velocity of the moving vehicle. Video Surveillance Systems is used to detect and track blobs. For each of those blobs we calculate the optical flows. RANSAC (RANdom SAmple Consensus) is employed to verify the best optical flow for each blob. A simple threshold approach is used to determine the most appropriate result of motion velocity.

***Keywords*—**motion tracking, velocity, moving vehicle, optical flow.

I. Introduction

OBJECT tracking has been intensely studied for its vast application area, which includes image segmentation, dynamic scene analysis, image registration, visual navigation, motion estimation and video compression.[1] One of the most popular approaches to track motion of object in video frame is to calculate a dense map of optical flows and then do segmenting or clustering on that map. However, most clustering algorithms need an initial number of groups or blobs that is obviously unknown. Other clustering algorithms that do not need the number of groups are so complex and costly to apply.

Optical flow algorithm has aperture problem that has to solve two unknowns in one equation. All optical flow methods try to introduce additional conditions so that we have more constrain to find the optical flow. Lucas-Kanade algorithm [2], a local least square calculation could be sufficient to compute optical flow sparsely for each blob. Another way of determining optical flow is in terms of discrete optimization. [3] The image matching is carried out using label assignment in the quantized search space, and the solution can be optimally found by minimizing the distance. Cross-correlation [4] can be a widely used block-based solution for the determination of optical flow. Sum of squared difference can be also used instead.

In this paper, iterative Lucas-Kanade algorithm [5] is implemented in pyramid scheme to compute optical flows of each blob and then RANSAC is exploited to figure out the best result of optical flow. Based on iterative Lucas-Kanade algorithm and RASAC, we come up with an integrated application that track blobs in frames of a video file and then apply computing optical flow for each blob so that we can determine velocity of the blobs' motion.

To insert images in *Word,* position the cursor at the insertion point and either use Insert | Picture | From File or copy the image to the Windows clipboard and then Edit | Paste Special | Picture (with "Float over text" unchecked).

Dong-min Woo is with Electronics Engineering Department, Myongji University, Gyeongjido, Korea 449-728. (e-mail: dmwoo2000@gmail.com).

An Nguyen Hong was with Electronics Engineering Department, Myongji University, Gyeongjido, Korea 449-728. (e-mail: hongantcnguyen @yahoo. com).

II. Existing Framework

A. Blob Tracking Modules of Video Surveillance Systems

The process of Video Surveillance System is described in Fig. 1. The system analyses frames in a video file and provides blobs information as output.

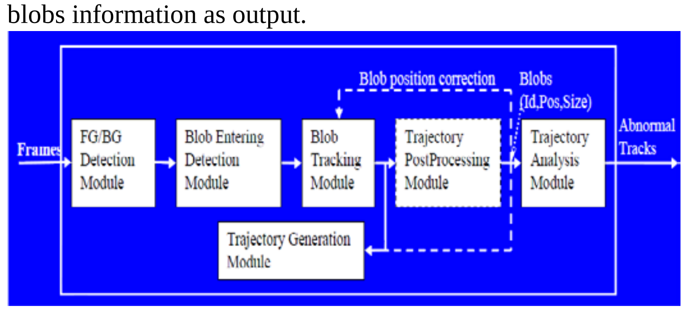

Fig. 1 Mapping nonlinear data to a higher dimensional feature space

B. Optical flow computation of Lucas –Kanade method

The Lucas- Kanade algorithm assumes optical flow is the same for all the pixels in a local neighborhood [2]. In addition, flow vectors must be small enough for the differential equation of optical flow to hold, which is often less than pixel spacing. For those vectors whose have larger magnitude, Lucas-Kanade-Tomasi algorithm with coarse-to-fine pyramid scheme is used to achieve a better accuracy.

III. Integrated Application: Tracking Motion and Calculating Velocity

Our application receives blobs from Video Surveillance system as input and processes more to have motion direction and velocity of each blob. Fig. 2. is an diagram of the process.

A. Detecting and Tracking Blob

In this first step, we have got the blobs by using Blob Tracking Module of Video Surveillance System. Instances of this implementation are shown in [6]. We can employ Kalman filtering for the estimation of blob position and size. [7] Each blob will be processed further to determine motion direction as well as motion velocity in next steps.

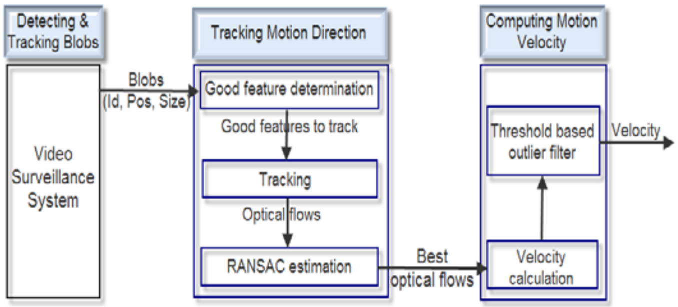

Fig. 2 Mapping nonlinear data to a higher dimensional feature space

B. Tracking Motion Direction

Determine a constant number of "Good features to track" for each blob.

For each blob, we define 5 points to track. Those 5 points firstly are selected by using Shi and Tomasi algorithm [8]. According to Shi and Tomasi, high eigenvalues imply texture and corner that is needed by a good feature. In other words, the minimum eigenvalue of spatial gradient matrix must be larger than a threshold. However, those good features are not always detected by the algorithm. Once we have number of detected features, we add more points so that we have enough 5 points. We simply determine 4 position points and center of a blob to add to the features to track.

Tracking "Good features to track" by using optical flow.

Optical flow is computed accurately by iterative Lucas-Kanade algorithm. Pyramid scheme increase robustness for computing.

RANSAC "RANdom SAmple Consensus" estimation of the best optical flow (direction and velocity) for each blob.

RANSAC is used to estimate the best model of data given a small set of inliers and outliers. Once we have the optical flows for each blob, we calculate angle and hypotenuse of each optical flow. Then we use RANSAC to determine the best model that optimally fits the flows. Through the RANSAC, we can eliminate noise from sparse calculating optical flow for 5 features to track.

C. Computing Motion Velocity

Using a threshold we find the most appropriate result of velocity. Noises between frames in a static camera's video file can come from many sources like camera's vibration, weather, or simply background noise. Even RANSAC can make mistake when refining from a bad source of optical flows of these noises. In providing a flexible solution to many kinds of video file, a threshold of magnitude of velocity is used to examine the best result from RANSAC process to eliminate outlier. Once the threshold is satisfied, the result is accepted.

IV. EXPERIMENT

Our application is developed based on Video Surveillance Systems and optical flow tracking system of OpenCV framework [9]. We tested our application with several highway camera video files without any knowledge of the camera parameter. Different video files with a variety of car intensity are used to test flexibility and stability of our application. In Fig. 3, a good quality video file with occlusion effect at separator is examined. In Fig. 4, a scene with a small truck turning right is examined. Fig. 5 is a scene with much more cars. And in Fig. 6, bad quality video file with less car intensity is examined. As it can be seen, the result could be very promising.

Fig. 3 Motion tracking and velocity estimation with occlusion effect at separator

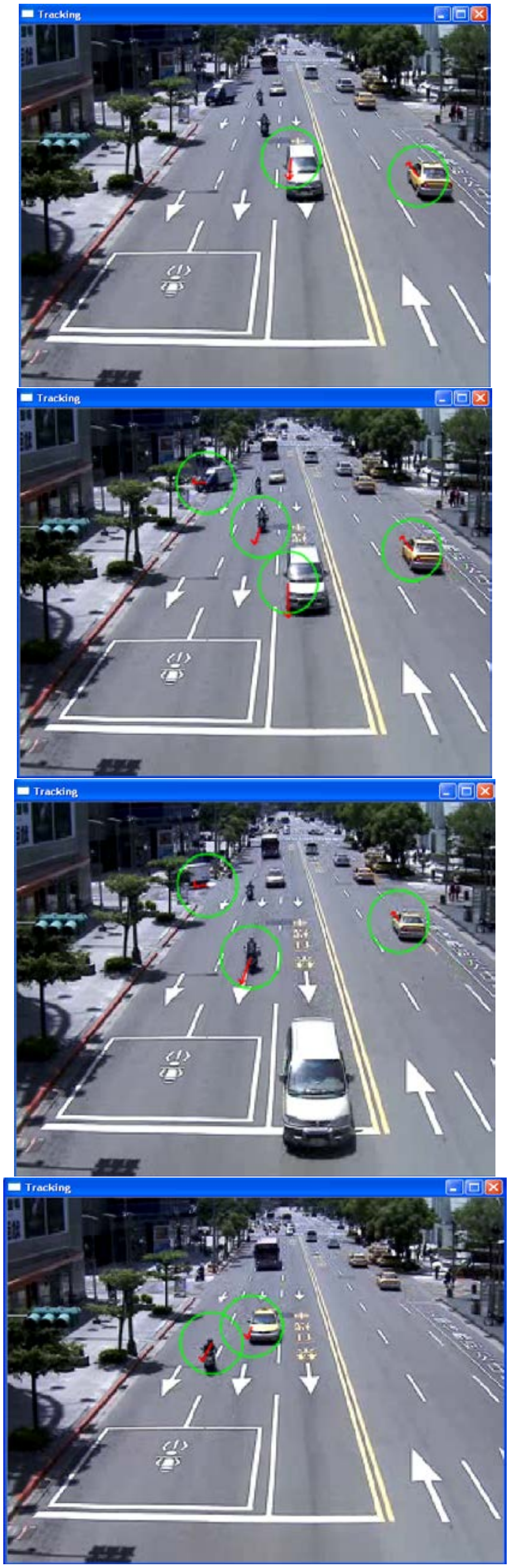

Fig. 4 Motion tracking and velocity estimation for a small trunk turning right

Fig. 5 Motion tracking and velocity estimation for many cars

Fig. 6 Motion tracking and velocity estimation in bad quality video file

V. CONCLUSION

In this paper, we have proposed a new procedure to track and determine velocity of blobs by integrating several strong tools like Video Surveillance Systems with optical flow computation. We also developed an application and test its performance with some different video files of highway traffic camera. The experiments of tracking moving vehicles are carried out from the slight traffic environment to heavy traffic environment. A threshold is applied manually at last to reduce outliers that do not come from real motion in the video. In future research, the proposed method can be improved by including autonomous learning and detection of the value of threshold in order to fit to motion in the video.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(No. NRF-2012R1A1A2004950).

REFERENCES

[1] C. S. Royden, K. D Moore, "Use of speed cues in the detection of moving objects by moving observers," Vision research, Vol. 59, pp. 17-24 2012. http://dx.doi.org/10.1016/j.visres.2012.02.006

[2] B. D. Lucas, T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," Proceeding of Imaging Understanding Workshop, pp. 121-130, 1981.

[3] B. Glocker, N. Komodakis, G. Tziritas, N. Navab, N. Paragios, "Dense Image Registration through MRFs and Efficient Linear Programming," Medical Image Analysis Journal, 2008.

[4] S. S. Beauchemin, J. L. Barron, *The computation of optical flow*, ACM New York, USA, 1995.

[5] Jean-Yves Bouguet, *Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm*, 2000.

[6] L. Li, W. Huang, I. Gu, Q. Tian 'Foreground object detection from videos containing complex background," Proceedings of the eleventh ACM international conference on Multimedia, pp. 2-10, 2003. http://dx.doi.org/10.1145/957013.957017

[7] Z. Zhu, J. Qiang, K. Fujimura, K. Lee, "*Combining Kalman filtering and mean shift for real time eye tracking under active IR illumination*," 16th International Conference on Pattern Recognition, pp. 318-321, 2002.

[8] J. Shi and C. Tomasi, "*Good features to track*," Proceeding 9th IEEE Conference on Computer Vision and Pattern Recognition, Springer, 1994.

[9] David Stavens, *The OpenCV Library: Computing Optical Flow*, Stanford Artificial Intelligence Lab, 2007.