**Contemporary Engineering Sciences, Vol. 9, 2016, no. 17, 843 - 851**
**HIKARI Ltd, www.m-hikari.com**
**http://dx.doi.org/10.12988/ces.2016.6696**

# Multi-Resolution Estimation of Optical Flow for Vehicle Tracking

**Viet Dung Do**

Image Processing Lab., Department of Electronics Engineering
Myongji University, Yongin, Republic of Korea

**Dong-Min Woo***

Department of Electronics Engineering, Myongji University
Yongin, Republic of Korea
*Corresponding author



## Abstract

This paper presents a hierarchical multi-resolution estimation of optical flow for a vehicle tracking system which can be used in a practical environment. Aiming at accurate estimation of optical flow, we construct a strong feature tracking system based on the Shi-Tomasi approach. As a feature detector, we use a Scale-Invariant Feature Transform (SIFT) algorithm, which not only firmly focuses on multi-scaling images but also correctly tracks strong interest points. The pyramidal Lucas-Kanade optical flow algorithm using our feature tracking system is then implemented. The information after estimating the optical flow can be used in the later tracking and detecting processes. For evaluation, we use the Autonomous Agents for On-Scene Networked Incident Management (ATON) project's highway video files, which include moving shadows. To test the tracking in a practical environment, we artificially add three adverse effects: additive Gaussian noise, vibration and motion blur. Experimental results demonstrate good performance for our multi-resolution optical flow system. The study confirms that the computed optical flow has a very small error with minimum eigenvalues in the optical flow equations.

**Keywords**: Vehicle tracking, Optical flow, Multi-resolution, Interest point

## 1 Introduction

Motion analysis has many application domains, including image segmentation, dynamic scene analysis, image registration, visual navigation, motion estimation and video compression. Recently, it has also been employed in robot and vehicle navigation [1]. This particular application is significantly important for autonomous car driving, obstacle detection and avoidance, and surveillance and monitoring of vehicles on the road. One of the most popular approaches to track motion of vehicles in a video frame is to calculate a dense map of optical flows. This study was motivated by the need for an efficient vehicle tracking system.

There has been a significant body of research on optical flow and a variety of algorithms have been developed for the accurate and rapid estimation of optical flow. All optical flow methods try to introduce additional conditions so that we have more flexible options in finding the optical flow. [2] The Lucas-Kanade algorithm [3], 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 [4]. 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 [5] [6] is a widely used block-based solution for determining optical flow. The sum of the squared difference can also be used instead.

One of the practical approaches for estimating optical flow is a hierarchical multi-resolution scheme. A multi-resolution problem is becoming more common for working with multi-scale and multi-view data. For instance, if there are high-speed cars, the motion or the displacement of these cars between two frames is relatively large. One feasible solution now is handling this problem at a lower resolution. The lower the resolution, the smaller the displacement, so the accuracy is improved. Experiments in [7] [8] showed promising results for correlation-based optical flow in noisy environments. The pyramid scheme can be used to implement the iterative Lucas-Kanade algorithm for the computation of optical flows of each blob. The pyramidal Lucas-Kanade optical flow algorithm also shows good performance for the vehicle tracking [9]. In this paper, we extend the pyramidal Lucas-Kanade algorithm to cope with a more practical environment by combining it with an efficient feature tracking scheme.

In order to reduce the computation time, optical flow algorithms should only be computed on interest points. An interest point is a well-defined position that can be detected with affine transformations, including translation, rotation and scaling. The Moravec corner detector [10] was one of the first detectors. This corner detector was then improved by Harris and Stephens [11]. The original Harris corner detector performed well on motion tracking. One efficient corner detector is the Scale-Invariant Feature Transform (SIFT) [12]. Features detected by SIFT are useful in solving the problem between multi-views of an object or a scene, since they are invariant to affine transformations even with the addition of noise and motion blur. In this study, after detecting strong interest points using SIFT, we applied the Shi and Tomasi method [13] to track good features.

## 2 Tracking Methodology

The suggested model for vehicle tracking consists of feature tracking and optical flow estimation. The overall process of our tracking system is described in Figure 1. Our application takes video files or sequences of images as input data. These inputs should be recorded by the highway traffic by a fixed camera. The video files that we used in our experiments have moving shadows from vehicles. This further challenged our testing.

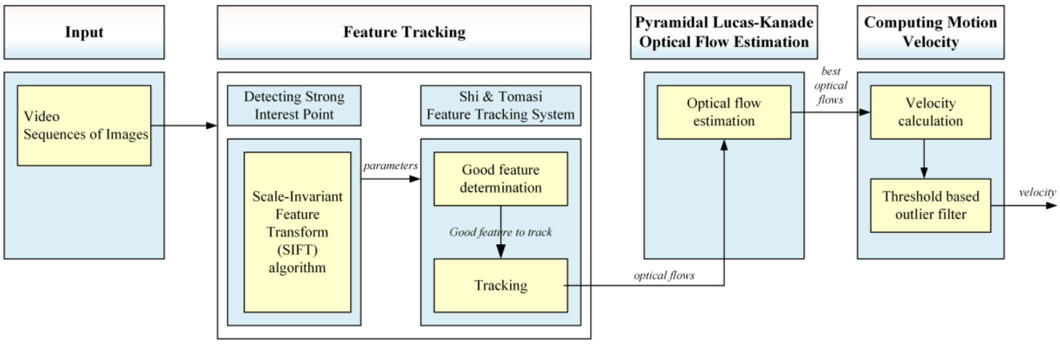


Fig. 1. Overall vehicle tracking system

For feature tracking, we employed the Scale-Invariant Feature Transform (SIFT) algorithm as a corner detector in order to detect strong interest points. After that, we used these points as parameters in the Shi-Tomasi feature tracking system. The optical flow estimation step takes the results from the previous step, and computes the optical flow. We implement the iterative Lucas-Kanade algorithm in combination with a pyramid scheme in order to solve the multi-resolution problem.

Our outputs are motion vectors of moving vehicles. By using these motion vectors, we can calculate other useful information such as vehicle detection, the velocity of vehicles, and the number of vehicles passing through in a period of time. This information is used later.

## 3 Feature Tracking

Our feature tracking system is based on the Shi-Tomasi algorithm. In order to solve a multi-resolution problem, we use the Scale-Invariant Feature Transform (SIFT) algorithm as a corner detector. Other corner detectors do not have good performance in matching images of different sizes because they are sensitive to scaling change. The SIFT algorithm is carried out as follows:

1. Given an input image $\boldsymbol{I(x,y)}$, compute the Difference-of-Gaussian function convolved with the image from the difference of two nearby scales separated by a constant multiplicative factor k.

$$D(x,y,\sigma) = \big(G(x,y,k\sigma) - G(x,y,\sigma)\big)I(x,y) \tag{1}$$

2. Compare each sample point to its eight neighbors in the current image and the neighbors in the scale above and below to detect the local maxima and minima of $\boldsymbol{D(x,y,\sigma)}$.

3. Calculate the location of the extremum by taking the derivative of the function with respect to x and setting it to zero.

$$\hat{x} = -\frac{\partial^2 D}{\partial x^2}^{-1} \frac{\partial D}{\partial x} \tag{2}$$

   where $D(\boldsymbol{x}) = D + \frac{\partial D}{\partial x}^{T} \boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^T \frac{\partial^2 D}{\partial x^2}\boldsymbol{x}$ is evaluated at the sample point and $\boldsymbol{x} = (x,y,\sigma)^T$ is the offset from this point.

4. Compute the principal curvature from a 2x2 Hessian matrix:

$$H = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{pmatrix} \tag{3}$$

5. Orientation assignment. For each image sample, $\boldsymbol{L(x,y)}$, compute the gradient magnitude, $\boldsymbol{m(x,y)}$, and orientation, $\boldsymbol{\theta(x,y)}$, using pixel differences.

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{4}$$

$$\theta(x,y) = tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right) \tag{5}$$

6. A 16x16 neighborhood around the keypoint is taken and divided into 16 sub-blocks of 4x4 sizes. For each sub-block, an 8-bin orientation histogram is created.
7. Match keypoints between two images by identifying their nearest neighbors.

Using the SIFT algorithm, we are able to obtain the very strong interest points. These interest points are used as parameters in the Shi-Tomasi feature tracking algorithm.

## 4 Optical Flow Estimation

### 4.1 Lucas-Kanade method

The Lucas-Kanade method is a famous differential method in optical flow estimation. In this method, we assume the velocity $v = (v_x, v_y)$ is constant over a small neighbourhood $\Omega_x$ of every $x \in \Omega$. The goal is for all *x*, so we need to minimize the following function:

$$\sum_{x \in \Omega} W^2(x)[\nabla I(x,t).v + I_t(x,t)]^2 \tag{6}$$

For *n* points $x \in \Omega$ at any *t*,

$$A = [\nabla I(x_1), \ldots, \nabla I(x_n)]^T \tag{7}$$

$$W = diag[W(x_1), \ldots, W(x_n)] \tag{8}$$

$$b = -(I_t(x_1), \ldots, I_t(x_n))^T \tag{9}$$

The solution is given by

$$A^T W^2 A v = A^T W^2 b \tag{10}$$

It is easy to implement the Lucas-Kanade algorithm. Since the algorithm has a fast calculation time with accurate calculation of derivatives, we used it in our study for its combination of accuracy and speed.

**4.2 Multi-Resolution Scheme**

The image pyramid is widely used as architecture to solve a multi-resolution problem. An image pyramid is a collection of images. All images are successively sampled from a single original image until some desired stopping point is reached. There are two types of image pyramid: the Gaussian and the Laplacian. The Gaussian pyramid is used to down-sample images, while the Laplacian pyramid is used to reconstruct an up-sample image from an image lower in the pyramid. Figure 2 shows three level pyramid.

One of the difficulties in implementing the Lucas-Kanade algorithm is that points might be moved outside the local window. For this reason, it is hard for the algorithm to find these points. This leads to the pyramidal Lucas-Kanade algorithm, which starts from the highest level of an image pyramid and continues to proceed to the lower levels. The pyramidal Lucas-Kanade optical flow algorithm is carried out as follows:

1. Given a height of the pyramid $L^m$, compute the optical flow at the deepest pyramid level $L^m$ using the original Lucas-Kanade optical flow algorithm.
2. Propagate the result to the upper level $L^{m-1}$ as an initial guess for the pixel displacement.
3. Compute the optical flow at the pyramid level $L^{m-1}$.
4. Repeat the same process until the highest pyramidal level is reached.

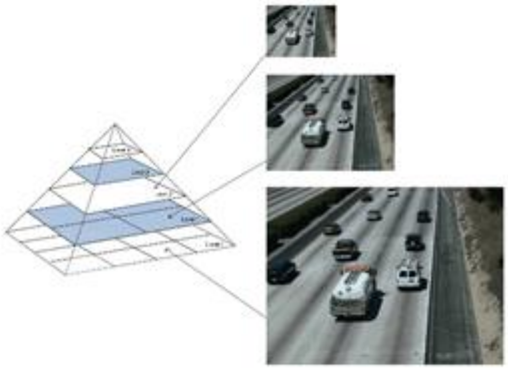

Fig. 2. Three level pyramid

## 5 Results and Discussion

We used two highway camera video files from the Autonomous Agents for On-Scene Networked Incident Management (ATON) project for the tracking experiment. The files also included the moving shadows of vehicles, which play an adverse role in the tracking process. The numbers of frames of these two files were 440 and 500.

The quality of video files is not always good. In order to test our application in a practical environment, we artificially applied three kinds of effect to the input video file: additive Gaussian noise, vibration and blurring. Figure 3, Figure 4 and Figure 5 show the tracking results of the first video (440 frames) with three kinds of effect: noise, vibration and blurring, respectively.


*frame 12 of 440* *frame 179 of 440* *frame 139 of 440* *frame 356 of 440*

Fig. 3. Vehicle tracking with additive Gaussian noise and additive Laplacian noise

*Frame 41 of 440* *Frame 78 of 440* *Frame 118 of 440* *Frame 365 of 440*

Fig. 4. Vehicle tracking with vibration


*Frame 13 of 440* *Frame 121 of 440* *Frame 141 of 440* *Frame 185 of 440*

Fig. 5. Vehicle tracking with blurring

The results show that our tracking system can efficiently track and detect all the moving vehicles in spite of all the adverse effects. However, there were small features from moving shadows which could not be eliminated. When we carry out the same experiments using the second video (550 frames), the tracking can be similarly carried out, regardless of the frame length.

We performed these experiments on Intel Core i5-3570 in a Microsoft Visual Studio 2010 C++ environment. Table 1 shows the computation time on two highway video files.

Table 1. Computational time of vehicle tracking

| Environment | Input | |
|---|---|---|
| | First video (440 frames) | Second video (500 frames) |
| Additive Gaussian Noise | 5.891 | 7.338 |
| Vibration Effect | 9.447 | 11.24 |
| Blurring Effect | 5.091 | 8.459 |

Our experimental results confirmed that our method has good performance with small errors in estimating optical flows, even under unfavorable environments. The accuracy of the first video file was better because the moving shadows in this input video were smaller than in the second video.

## 6 Conclusion

In this paper, we proposed a hierarchical multi-resolution estimation method of optical flow. Our vehicle tracking system method can be used effectively in practical environments. To verify the effectiveness of our method, we carried out experiments with additive Gaussian noise, vibration and blurring. Our experimental results confirmed that the proposed method can be effectively used in tracking of moving vehicles in practical environments. As an extension of this work, we need to solve more complex problems in motion tracking, including occlusion of objects and lighting variation.

**Acknowledgements**. 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] S. Kim, S. Lee, S. Kim, J. Lee, Object tracking of mobile robot using moving color and shape information for the aged walking, *Int. Journal of Advanced Science and Technology*, **3** (2009), 59-68.

[2] D. Woo, Optical flow clustering using centroid neural network for motion tracking of moving vehicles, *Journal of Computers*, **10** (2015), no. 3, 213-220. http://dx.doi.org/10.17706/jcp.10.3.213-220

[3] B. D. Lucas, T. Kanade, An Iterative image registration technique with an application to stereo vision, *Proc. of Imaging Understanding Workshop*, (1981), 121-130.

[4] B. Glocker, N. Komodakis, G. Tziritas, N. Navab, N. Paragios, Dense image registration through MRFs and effficient linear programming, *Medical Image Analysis*, **12** (2008), no. 6, 731-741. http://dx.doi.org/10.1016/j.media.2008.03.006

[5] S. S. Beauchemin, J. L. Barron, The computation of optical flow, *ACM Computing Surveys*, **27** (1995), no. 3, 433-466. http://dx.doi.org/10.1145/212094.212141

[6] M. I. Khan, J. Ahmed, A. Ali, A. Masood, Robust edge-enhanced fragment based normalized correlation tracking in cluttered and occluded imagery, *Int. Journal of Advanced Science and Technology*, **12** (2009), 25-34.

[7] D. Kesrarat, V. Patanavijit, Experimental efficiency analysis in robust models of spatial correlation optical flow methods under non Gaussian noisy contamination, *Proc. of ECTI-CON*, (2013), 1-6. http://dx.doi.org/10.1109/ecticon.2013.6559474

[8] T. Lee, D. Anderson, Performance analysis of a correlation-based optical flow algorithm under noisy environments, *Proc. of IEEE International Symposium on Circuits and Systems*, (2006). http://dx.doi.org/10.1109/iscas.2006.1693679

[9] V. Do, D. Woo, Multi-resolution estimation of optical flow on vehicle tracking under unpredictable environments, *Advanced Science and Technology Letters*, **129** (2016), 32-36.

http://dx.doi.org/10.14257/astl.2016.129.07

[10] C. Schmid, R. Mohr, Local grayvalue invariants for image retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19** (1997), no. 5, 530-534. http://dx.doi.org/10.1109/34.589215

[11] C. Harris, M. Stephens, A combined corner and edge detector, *Proc. of Fourth Alvey Vision Conference*, (1988), 147-151. http://dx.doi.org/10.5244/c.2.23

[12] David G. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, **60** (2004), no. 2, 91-110. http://dx.doi.org/10.1023/b:visi.0000029664.99615.94

[13] J. Shi, C. Tomasi, Good features to track, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94*, (1994), 593-600. http://dx.doi.org/10.1109/cvpr.1994.323794

**Received: April 15, 2016; Published: August 11, 2016**