

Detection of Moving Vehicles Using ANN-Based Clustering of Optical Flow

Dong-Min Woo¹

Abstract—This paper presents the detection of moving vehicles using clustering of optical flow. ANN (Artificial Neural Network) algorithms are employed to group pixels of moving vehicles from traffic images based on the clustering of optical flow. For the estimation of optical flow, 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. To verify the efficiency of optical flow clustering, we use the Autonomous Agents for On-Scene Networked Incident Management (ATON) project's highway video files. Experimental results demonstrate good performance for the segmentation of moving vehicles. The study confirms that the proposed method can efficiently detect moving vehicles out of background and accurately estimate the velocity of moving vehicle.

Keywords—moving vehicles, optical flow, ANN (Artificial Neural Network), clustering, feature.

I. INTRODUCTION

MOTION analysis is significantly important for autonomous car driving, obstacle detection and avoidance, and surveillance and monitoring of vehicles on the road. [1] 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 detection by using optical flow.

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. The pyramid scheme [5], [6] can be used to implement the iterative Lucas-Kanade algorithm for the computation of optical flows of each blob.

In this paper, we are concerned with a grouping process of

the optical flows in terms of ANN clustering algorithm. This study is the extension of the previous work [7] by using the central neural network (CNN) algorithm. Segmentation of moving vehicles results from this grouping process in terms of CNN, since it has advantageous features such as requiring neither a predetermined schedule for learning coefficients nor the total number of iterations for clustering. However, it needs the proper adjustment of threshold for the reduction of outlier occurrence. In this context, FCM (Fuzzy C-mean) and GBFCM (Gradient Based Fuzzy C-mean) clustering algorithms are employed to implement the efficient clustering of optical flow, since these ANN-based clustering algorithms are known as robust in this particular situation.

II. OPTICAL FLOW ESTIMATION

A. 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 Ω_x of every $x \in \Omega$. The goal is for all x , so we need to minimize the following function:

$$\sum_{x \in \Omega} W^2(x) [\nabla I(x, t) \cdot v + I_t(x, t)]^2 \quad (1)$$

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

$$A = [\nabla I(x_1), \dots, \nabla I(x_n)]^T \quad (2)$$

$$W = \text{diag}[W(x_1), \dots, W(x_n)] \quad (3)$$

$$b = -(I_t(x_1), \dots, I_t(x_n))^T \quad (4)$$

The solution is given by

$$A^T W^2 A v = A^T W^2 b \quad (5)$$

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.

B. 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

Dong-Min Woo¹ is with Myongji University, Yongin, Gyeonggido 17058, KOREA (corresponding author's phone: +82-31-330-6752; e-mail: dmwoo@mju.ac.kr).

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. Fig. 1 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.

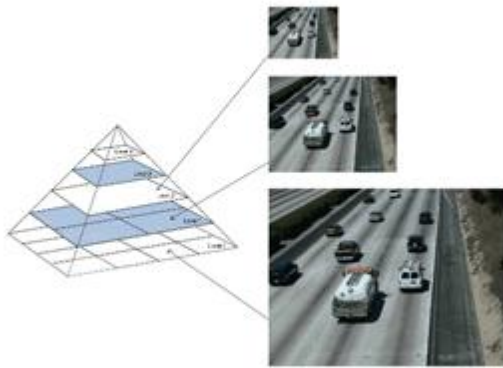


Fig.1 Three-level pyramid.

III. ANN CLUSTERING ALGORITHMS

A. Fuzzy c -Means (FCM) Algorithm

The objective of clustering algorithms is to group of similar objects and separate dissimilar ones. Bezdek first generalized the fuzzy ISODATA by defining a family of objective functions J_m , $1 < m < \infty$, and established a convergence theorem for that family of objective functions [8], [9]. For FCM, the objective function is defined as:

$$J_m(U, v) = \sum_{k=1}^n \sum_{i=1}^c (\mu_{ki})^m (d_i(x_k))^2 \tag{6}$$

where $d_i(x_k)$ denotes the distance from the input data x_k to v_i , the center of the cluster i , μ_{ki} is the membership value of the data x_k to the cluster i , and m is the weighting

exponent, $m \in 1, \dots, \infty$ while n and c are the number of input data and clusters, respectively. Note that the distance measure used in FCM is the Euclidean distance.

Bezdek defined a condition for minimizing the objective function with the following two equations:

$$\mu_{ki} = \frac{1}{\sum_{j=1}^c \left(\frac{d_i(x_k)}{d_j(x_k)}\right)^{\frac{2}{m-1}}} \tag{7}$$

$$v_i = \frac{\sum_{k=1}^n (\mu_{ki})^m x_k}{\sum_{k=1}^n (\mu_{ki})^m} \tag{8}$$

The FCM finds the optimal values of group centers iteratively by applying (7) and (8) in an alternating fashion.

B. Gradient-Based Fuzzy c -Means (GBFCM) Algorithm

One attempt to improve the FCM algorithm was made by minimizing the objective function using one input data at a time instead of the entire input data. That is, the FCM in (7) and (8) uses all data to update the center value of the cluster, but the GBFCM that is used in this paper was developed to update the center value of the cluster with a given individual data sequentially [10]. Given one datum x_k and c clusters with centers at v_j , ($j = 1, 2, \dots, c$), the objective function to be minimized is:

$$J_k = \mu_{k1}^2 (v_1 - x_k)^2 + \mu_{k2}^2 (v_2 - x_k)^2 + \dots + \mu_{kc}^2 (v_c - x_k)^2 \tag{9}$$

with the following constraint:

$$\mu_{k1} + \mu_{k2} + \dots + \mu_{kc} = 1 \tag{10}$$

The basic procedure of the gradient descent method is that starting from an initial center vector, $v_i(0)$, the gradient ΔJ_k of the current objective function can be computed. The next value of v_i is obtained by moving to the direction of the negative gradient along the error surface such that:

$$v_i(n+1) = v_i(n) - \eta \frac{\partial J_k}{\partial v_i(n)} \tag{11}$$

where η is the iteration index and

$$\frac{\partial J_k}{\partial v_i(n)} = 2\mu_{ki}^2 (v_i(n) - x_k) \tag{12}$$

Equivalently,

$$v_i(n+1) = v_i(n) - 2\eta\mu_{ki}^2 (v_i(n) - x_k) \tag{13}$$

where η is a learning constant.

A necessary condition for optimal positions of the centers for the groups can be found by the following:

$$\frac{\partial J_k}{\partial \mu} = 0 \tag{14}$$

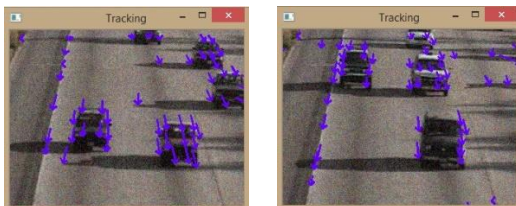
After applying the condition of (14), the membership grades can be found as:

$$\mu_{ki} = \frac{1}{\sum_{j=1}^c \left(\frac{d_i(x_k)}{d_j(x_k)} \right)^2} \quad (15)$$

Both the FCM and GBFCM have an objective function that related the distance between each center and data with a membership grade reflecting the degree of their similarities with respect to other centers. On the other hand, they differ in the way they try to minimize it.

IV. EXPERIMENTAL RESULTS

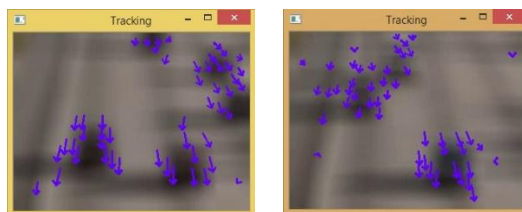
We used two highway camera video files from the Autonomous Agents for On-Scene Networked Incident Management (ATON) project for the tracking experiment. The files also included the moving shadows of vehicles, which play an adverse role in the tracking process. The quality of video files is not always good. In order to test our application in a practical environment, we artificially applied two kinds of effect to the input video file: additive Gaussian noise and blurring. Fig. 2 and Fig. 3 show estimated optical flow of the traffic video (440 frames) with two kinds of effect: noise and blurring, respectively.



frame 12 of 440

frame 179 of 440

Fig. 2 Vehicle tracking with additive Gaussian noise and additive Laplacian noise



Frame 13 of 440

Frame 121 of 440

Fig. 3 Vehicle tracking with blurring

CNN-based clustering algorithm detects moving vehicle in this unfavourable situation. However, without the proper adjustment of threshold, many outliers have occurred. To carry out vehicle detection experiments, we implemented two ANN-based classifiers: FCM and GBFCM.

For various occasions, shown in Fig. 3 and 4, both of our ANN-based clustering algorithm detects each blob of moving vehicles. Even though some moving vehicles are not detected in an unfavourable situation, most vehicles are successfully detected by using our clustering algorithms. Considering real

world difficulty of tracking vehicles, this experimental result is very promising.

Especially in Fig. 3, the image files in the video file are not very clear with blurring. Moreover, there is a fog, which significantly obscures the detection procedure. The tracking results show that our CNN-based clustering algorithm did not detect moving vehicle in this unfavourable situation. In terms of the experimental results, we are convinced that our tracking method using ANN-based clustering algorithms can be efficiently applied to the monitoring of traffic information.

V. CONCLUSION

In this paper, we have proposed a new procedure to track and detect the moving vehicles by clustering optical flows. ANN-based clustering algorithms, FCM and GBFCM, are employed to group pixels of moving vehicles from traffic images. 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. In future research, the proposed method can be improved by extracting various kinds of traffic information from video data.

REFERENCES

- [1] C. S. Royden, "Use of speed cues in the detection of moving objects by moving observers," *Vision Research*, vol. 59, 2012, pp. 17–24.
- [2] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Imaging Understanding Workshop*, 1981, pp. 121–130.
- [3] B. Clocker, N. Komodakis, G. Tziritas, N. Navab and N. Paragios, "Dense Image Registration through MRFs and Efficient Linear Programming," *Medical Image Analysis Journal*, 2008.
- [4] S. S. Beauchemin and J. L. Barron, *The computation of optical flow*, ACM New York, USA, 1995.
- [5] D. M. Woo and A. N. Hong, "An integrated solution to motion tracking of moving vehicles using optical flow," *Int. Journal of Computing, Communication and Instrumentation Eng.*, vol. 1, no. 1, 2014, pp. 60-63.
- [6] D. V. Dung and D. M. Woo, "Multi-Resolution Estimation of Optical Flow for Vehicle Tracking," *Contemporary Engineering Sciences*, vol. 9, no. 17, 2016, pp. 843-851.
- [7] D. M. Woo, "Optical Flow Clustering Using Centroid Neural Network for Motion Tracking of Moving Vehicles," *Journal of Computers*, vol. 10, no. 3, 2015, pp. 213-220.
- [8] J. C. Bezdek, "A convergence theorem for the fuzzy ISODATA clustering algorithms," *IEEE Trans. Pattern Anal. Mach. Int.*, vol. 2, 1980, pp. 1-8.
- [9] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*, New York : Plenum, 1981.
- [10] C. Looney, *Pattern Recognition Using Neural Networks*, New York, Oxford University press, 1997, pp. 252-254.