Motion-Vector Clustering for Traffic Speed Detection From UAV Video

Ruimin Ke
Department of Civil and
Environmental Engineering
University of Washington
Seattle, WA, U.S.A.
Email: ker27@uw.edu

Sung Kim
Department of
Electrical Engineering
University of Washington
Seattle, WA, U.S.A.
Email: sungk9@uw.edu

Zhibin Li
Department of Civil and
Environmental Engineering
University of Washington
Seattle, WA, U.S.A.
Email:lizhibin@uw.edu

Yinhai Wang
Department of Civil and
Environmental Engineering
University of Washington
Seattle, WA, U.S.A.
Telephone: 206-616-2696
Email:yinhai@uw.edu

***Abstract*—A novel method for detecting the average speed of traffic from non-stationary aerial video is presented. The method first extracts interest points from a pair of frames and performs interest point tracking with an optical flow algorithm. The output of the optical flow is a set of motion vectors which are k-means clustered in velocity space. The centers of the clusters correspond to the average velocities of traffic and the background, and are used to determine the speed of traffic relative to the background. The proposed method is tested on a 70-frame test sequence of UAV aerial video, and achieves an average error for speed estimates of less than 12%.**

***Keywords*—Traffic speed detection, unmanned aerial vehicle (UAV), optical flow, interest point tracking, k-means clustering.**

I. Introduction

Traffic monitoring technologies provide information that is critical to improving the efficiency and safety of roadway transportation. For instance, information on traffic flow rate or vehicle density can be used to mitigate roadway congestion by dynamically routing vehicles away from congested areas. In general, intelligent transportation systems (ITS) have the potential to improve transportation delays, roadway safety, and energy efficiency [1].

Compared to traditional methods, roadway monitoring with unmanned aerial vehicles (UAV's) has several advantages. Traditional methods of roadway surveillance include in-roadway sensors (e.g., pneumatic road tubes or inductive loops), as well as over-roadway sensors such as fixed video cameras [2]. In contrast to the traditional surveillance methods mentioned above, UAV's can both monitor a large continuous stretch of roadway, and focus in on a specific road segment. Fixed cameras need to be placed in a dense configuration in order to have similar surveillance coverage, which may not be cost effective [3] [4]. For roadway incidents, UAV's can be used to provide rapid assessment and reconnaissance of the incident site for emergency response teams [4]. For data collection, aerial footage taken by UAV's can be used to perform vehicle tracking, as well as collect a variety of statistics such as traffic density, average annual daily travel, and estimated travel time [5].

In this work, we propose a method to detect the average speed of traffic streams from aerial UAV footage, where a traffic stream is defined as a continuous sequence of vehicles traveling with similar speed and direction. Specifically, we extract Shi-Tomasi interest points [6] from a region of interest (ROI) in a pair of frames, and extract the motion of the interest points with a pyramidal Kanade-Lucas algorithm for optical flow [7] [8]. With the motion vectors from the optical flow algorithm as input, we cluster the interest points in velocity space to identify the cluster centers. Since vehicles in a traffic stream generally have similar speed and direction, the centers of the velocity clusters naturally correspond to the average speed of the traffic stream.

The remainder of the paper is organized as follows. Section II describes previous work related to traffic analysis and vehicle detection with UAV aerial footage. The details of the proposed method for speed detection is covered in Section III, and includes discussions of interest point extraction, optical flow, and k-means clustering. Experimental results for the proposed method are presented in Section IV, and we conclude with a summary of the paper and a discussion of future work in Section V.

II. Related Work

Extracting traffic information from UAV videos is an area of active research in transportation engineering. For instance, recent work has included road detection and incident detection [5] [9], AADT analysis [4], and vehicle speed detection [10] [13]. Moranduzzo et. al. developed an approach for vehicle speed detection by finding correspondence between SIFT (Scale-Invariant Feature Transform) features across video frames [10]. Similarly, Zhang et. al. used SIFT with RANSAC (Random Sample Consensus) to deal with image registration, and a separate process to perform vehicle tracking and speed estimation.

While the approaches described above satisfy correctness, brute force methods for feature correspondence are inefficient and scale in at least $O(n^2)$ time with the number of features. In comparison, we should expect a guided search to have better performance and scalability. In our approach, we apply an iterative optical flow technique that makes use of the spatial intensity gradient of the image to guide the correspondence search [7]. Optical flow-based speed detection has been applied successfully in the past when applied to cases where cameras are fixed [11] [12]. As for UAV videos, only one study yet successfully extracts traffic speed making use of optical flow [14]. However, image registration is still the main idea in handling camera motion. The processing speed of this method is several seconds per frame, which is still very computationally expensive. To our best knowledge, our work is the first one that combines optical flow with clustering method in UAV based traffic speed detection, and our method is very efficient.

III. Framework for speed Detection

The framework for the proposed speed detection method can be segmented into two separate steps, which are described in the sub-sections below. In the first step, interest points are identified in a pair of consecutive frames; we use interest points where the eigenvalues of the second-moment matrix are large, i. e., Shi-Tomasi features. The Kanade-Lucas optical flow algorithm is then used to track interest points between the frame pair. In the second step, the speed and direction of interest points on both vehicles and the background are used as input to a clustering algorithm. Once the clusters have been identified, we can extract the average motion of traffic streams by subtracting the means of traffic clusters from the cluster center corresponding to the background.

A. Interest Point Tracking

Selecting good features is critical for tracking features robustly across image frames. To select Shi-Tomasi "good features," let the matrix

$$\mathbf{G} = \sum_{p_x-w_x}^{p_x+w_x} \sum_{p_y-w_y}^{p_y+w_y} \begin{bmatrix} \mathbf{I}_x^2 & \mathbf{I}_x\mathbf{I}_y \\ \mathbf{I}_x\mathbf{I}_y & \mathbf{I}_y^2 \end{bmatrix} \tag{1}$$

be the second-moment matrix of image $\mathbf{I}$ about point $\mathbf{u} = (p_x, p_y)$ in the window w of size $(2w_x + 1) \times (2w_y + 1)$. Then the interest points in $\mathbf{I}$ are located at the points $\mathbf{u}_i$ where $\mathbf{G}$ is non-singular, and the minimum eigenvalue $\lambda_{min} = \min(\lambda_1, \lambda_2)$ of $\mathbf{G}$ is above a specific threshold λ_{th}. To provide non-maximal suppression, any interest point $\mathbf{u}_i$ is not considered if there is another interest point $\mathbf{u'}_i$ in a 3x3 neighborhood about $\mathbf{u}_i$ with a larger λ_{min}. Finally, any interest points after the first n, sorted in order of decreasing λ_{min}, are not considered.

After interest points in frames $\mathbf{I}(x, y, t)$ and $\mathbf{I}(x, y, t + 1)$ have been identified, an interest point $\mathbf{u}_i$ can be tracked from time t to $t + 1$ with the Kanade-Lucas algorithm for optical flow. In order to track points across distances on the order of several pixels, we use an iterative implementation with image pyramids [8]; the method is summarized below.

Let $\mathbf{I}^L$ be the pyramidal image of $\mathbf{I}$ at pyramid level L. Then $\mathbf{I}^L$ is related to the original image $\mathbf{I}$ by the relation

$$\mathbf{u}^L = \frac{\mathbf{u}}{2^L} \tag{2}$$

where $\mathbf{u}$ is any point in $\mathbf{I}$. The objective is to find the optimal displacement $\boldsymbol{s}^*$ such that the error function $\epsilon(s_x, s_y)$ is minimized, that is,

$$\boldsymbol{s}^* = \underset{s}{\operatorname{argmin}} \left[\epsilon(s_x, s_y)\right], \tag{3}$$

where the error function

$$\epsilon(s_x, s_y) = \sum_{p_x-w_x}^{p_x+w_x} \sum_{p_y-w_y}^{p_y+w_y} \left(\mathbf{I}^L(p_x, p_y) - \mathbf{J}^L(p_x + s_x, p_y + s_y)\right)^2 \tag{4}$$

is the windowed sum-of-squared differences between images $\mathbf{I}^L$ and $\mathbf{J}^L = \mathbf{I}^L(t + 1)$.

If $\boldsymbol{s}^{L+1} = \mathbf{g}^{L+1} + \boldsymbol{\nu}^K$ is an approximation for $\boldsymbol{s}^*$ at layer $L + 1$, then the initial guess for the displacement $\mathbf{s}^L$ is $\bar{\boldsymbol{s}}\ ^0 = \mathbf{g}^L + \boldsymbol{\nu}^0$, where

$$\boldsymbol{\nu}^0 = [0\ 0]^\intercal,$$

$$\mathbf{g}^L = 2\mathbf{s}^{L+1} = 2(\mathbf{g}^{L+1} + \boldsymbol{\nu}^K). \tag{5}$$

The update rule for $\boldsymbol{\nu}$ is based on the first-order Taylor approximation for the partial derivative of $\epsilon(s_x, s_y)$ [8]. For iteration k,

$$\boldsymbol{\nu}^k = \boldsymbol{\nu}^{k-1} + \boldsymbol{\eta}^k, \tag{6}$$

where

$$\boldsymbol{\eta}^k = \mathbf{G}^{-1}\overline{\mathbf{b}}\ ^k, \tag{7}$$

$$\overline{\mathbf{b}}\ ^k = \sum_{p_x-w_x}^{p_x+w_x} \sum_{p_y-w_y}^{p_y+w_y} \begin{bmatrix} \boldsymbol{\delta}^k(p_x, p_y)\mathbf{I}_x^L(p_x, p_y) \\ \boldsymbol{\delta}^k(p_x, p_y)\mathbf{I}_y^L(p_x, p_y) \end{bmatrix}. \tag{8}$$

$\boldsymbol{\delta}^k$ is the difference between the images $\mathbf{I}$ and $\mathbf{J}$ for the displacement $\overline{\mathrm{s}}\ ^{k-1}$, given by

$$\begin{aligned} \boldsymbol{\delta}^k(p_x, p_y) &= \mathbf{I}^L(p_x, p_y) - \mathbf{J}^L(p_x + \overline{s}_x^{\ k-1}, p_y + \overline{s}_y^{\ k-1}) \\ &= \mathbf{I}^L(p_x, p_y) - \mathbf{J}^L(p_x + g_x^L + \nu_x^{k-1}, p_y + g_y^L + \nu_y^{k-1}). \end{aligned} \tag{9}$$

The iteration terminates either when $k > K - 1$, or $||\boldsymbol{\eta}^k||$ is less than a specified threshold. The iterative process described above is executed for each layer $L \in [0, L_m]$, starting from the image-layer L_m with the guess

$$\mathbf{g}^{L_m} = [0\ 0]^\intercal.$$

Hence, for the layer $L = 0$ corresponding to the original image $\mathbf{I}$, an interest point $\mathbf{u}$ can be found at the point $\mathbf{u} + \mathbf{s}^0$ in $\mathbf{J}$.

B. Velocity Clustering

The result of the optical flow described in the section above is a set of vectors V with elements $\mathbf{v_i} = (l_i, \theta_i)$, where l and θ are given by

$$\begin{aligned} l &= \sqrt{s_x^2 + s_y^2} \\ \theta &= \arctan(s_y, s_x) \\ \mathbf{s}^0 &= (s_x, s_y). \end{aligned}$$

Each element in V corresponds to an interest point $\mathbf{u_i}$ tracked from t to $t+1$, where $\mathbf{s}^0$ is the displacement for $\mathbf{u}_i$ calculated from the optical flow. Given V, its elements can be clustered with respect to l and θ using a standard k-means algorithm. For k traffic streams, the number of clusters should be set to $k+1$ to account for interest points in the background.

In order for clusters of interest points from traffic streams and the background to be correctly separated, both the background, and vehicles in a given traffic stream, should satisfy a ”similar motion” criteria. That is, interest points from the background should have small variation in l and θ between interest points. Likewise, for traffic clusters the variation in motion for vehicles within a given traffic stream must be small. Intuitively, if the motion criteria is violated, the clusters corresponding to different traffic streams, or between a traffic stream and the background, may become mixed. In practice, the motion criteria could be violated by a variety of conditions such as weather, or irregular vehicle movement caused by roadway obstructions.

C. Average speed Determination

Suppose there are k clusters corresponding to traffic steams and a single background cluster. If the cluster centers of the traffic streams are $\mathbf{v}_{i,t},\ i \in [1, k]$ and the cluster center of the background is $\mathbf{v}_{bg}$, then the average velocities of the traffic streams $\mathbf{v}_{i,avg}$ relative to the background are given by

$$\begin{aligned} \mathbf{v}_{i,avg} &= (l_{i,avg}, \theta_{i,avg}), \\ \theta_{i,avg} &= \arctan\left(d_y, d_x\right) \\ l_{i,avg} &= \sqrt{d_x^2 + d_y^2} \\ d_x &= l_{i,t}\cos\theta_{i,t} - l_{bg}\cos\theta_{bg} \\ d_y &= l_{i,t}\sin\theta_{i,t} - l_{bg}\sin\theta_{bg}. \end{aligned} \tag{10}$$

To convert distance in pixels to speed in kilometers per hour, we use the video frame rate and reference markings from video frames. For a video with frame rate f, and the pixel length and actual length of a reference mark are l_p and l_a, respectively, the actual speed s of a vehicle that moves d_p in one frame pair is determined by

$$s = \left(\frac{l_a}{l_p}\right) \times d_p / f \tag{11}$$

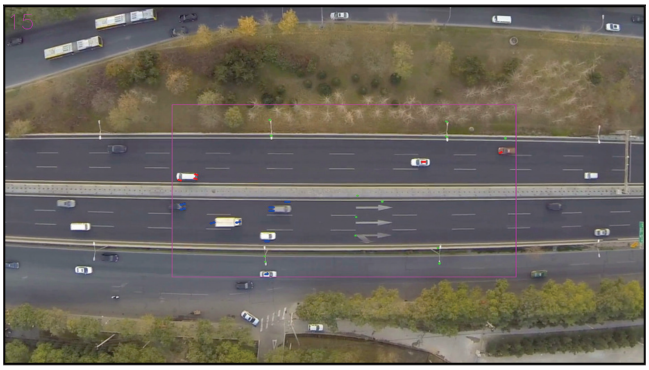

Fig. 1. The ROI is defined as a fixed rectangle in each frame, and is selected to exclude all traffic streams except those under analysis.

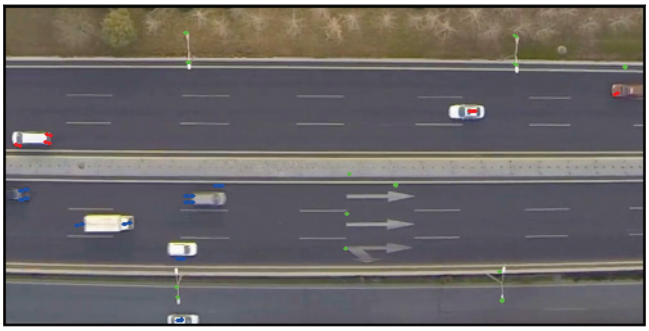

Fig. 2. A zoomed picture of traffic and the background with interest point detections, colored to show cluster membership. Traffic traveling left and right correspond to *Direction A* and *Direction B*, respectively.

IV. Results

The interest point tracking and velocity clustering process is implemented with C++ and OpenCV 2.4.11 [15]. The test dataset consists of a 70-frame video at 960×540 resolution, taken at 24 frames-per-second by a UAV traveling above a freeway segment. The ROI is selected to include 6 lanes of traffic moving in two directions, denoted *Direction A* and *Direction B*, resulting in a window of 500×250 pixels (Fig. 1). For the reference mark, we use lane markings with a measured pixel length and actual length of 36 pixels and 6 meters, respectively.

The average traffic speed for frame pairs is detected and then averaged over intervals of five consecutive frame pairs (Table I). Actual traffic speed for a direction is calculated by averaging the speed of every vehicle in that direction (Table II); the speed of individual vehicles is measured manually with an on-screen pixel measure.

Figure 4 shows the errors of the 5-frame average over 70 frames for both traffic directions. The average errors for the two traffic directions over the entire dataset are 11.695% and 10.956% (Table III).

Plots of the clustering results illustrate the separation of interest points in velocity space, and clusters are cleanly separated throughout testing (Fig. 3 and Fig. 2). From visual inspection, the most likely sources of error are (1) points from the background occasionally become mixed into clusters of interest points from traffic, and (2) vehicles with more

TABLE I. PREDICTED TRAFFIC SPEEDS (PIXELS/FRAME).

	F. 1-5	F. 6-10	F. 11-15	F. 16-20	F. 21-25
Direction A	5.839	5.758	5.704	5.767	5.763
Direction B	3.878	3.808	3.975	4.059	4.049
	F. 26-30	**F. 31-35**	**F. 36-40**	**F. 41-45**	**F. 46-50**
Direction A	5.896	6.007	6.097	6.004	5.949
Direction B	4.167	4.108	4.924	4.934	4.959
	F. 51-55	**F. 56-60**	**F. 61-65**	**F. 66-70**	**Avg.**
Direction A	6.082	6.615	6.533	6.341	6.025
Direction B	4.942	4.092	4.147	4.132	4.298

TABLE II. ACTUAL TRAFFIC SPEEDS (PIXELS/FRAME).

	F. 1-5	F. 6-10	F. 11-15	F. 16-20	F. 21-25
Direction A	4.800	4.900	5.733	5.667	4.933
Direction B	4.550	4.800	4.800	4.800	4.600
	F. 26-30	**F. 31-35**	**F. 36-40**	**F. 41-45**	**F. 46-50**
Direction A	5.100	4.700	5.500	5.650	5.580
Direction B	4.800	4.800	4.500	4.550	4.638
	F. 51-55	**F. 56-60**	**F. 61-65**	**F. 66-70**	**Avg.**
Direction A	5.660	5.800	5.975	5.900	5.421
Direction B	4.210	4.173	4.173	4.091	4.535

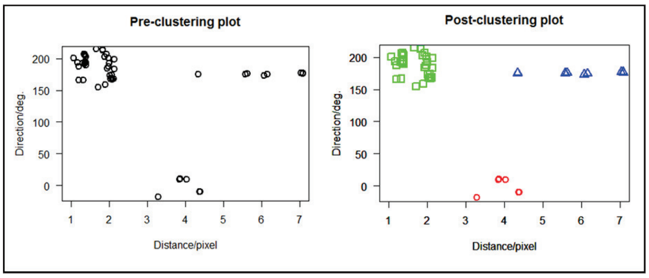

Fig. 3. Plots of the k-means input (left) and output (right) for the test frame shown in Fig. 2.

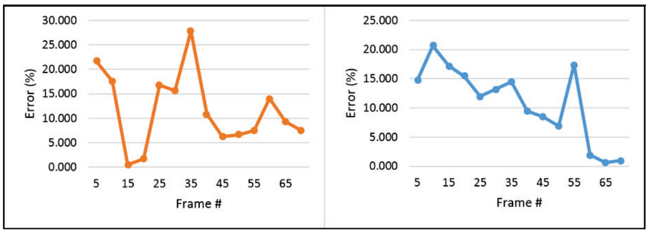

Fig. 4. Error for the 5-frame averages over the dataset for Direction B (left) and Direction A (right).

interest points are weighted more heavily in the average speed calculation.

TABLE III. SUMMARY OF RESULTS

	Predicted speed (km/h)	Actual speed (km/h)	Error(%)
Direction A	86.760	78.066	11.695
Direction B	61.891	65.304	10.956

V. CONCLUSION AND FUTURE WORK

In this work we propose a method to detect the average speed of traffic streams from UAV aerial video. The method can be divided into two steps, one where interest points are detected and tracked across frames, and another where the motion of interest points is clustered. For the interest point detection and tracking, we extract Shi-Tomasi interest points and track with an iterative optical flow algorithm. Using the optical flow vectors as input, the vectors are k-means clustered based on their speed and direction. The average traffic speed for a traffic stream can be determined by taking the vector difference between the means of the traffic cluster and the cluster corresponding to the background. Experiments on a dataset of 70 frames, taken by a UAV over a segment of freeway, show that the proposed method yields good results with average error less than 12%.

There are several directions for further work, in addition to improvements that would make the proposed method more amenable for use in a real system. We mention in Section III that the motion criteria depends on similar movement of both background and traffic interest points, but we were unable to test and quantify this relationship. In general, the method could also be tested under a wider set of conditions, such as varying weather or time-of-day. At the system level, this work could be combined with methods for automatic reference mark detection and traffic lane detection.

ACKNOWLEDGMENT

The authors would like to thank Beihang University, China for providing the UAV video dataset.

REFERENCES

[1] L. Figueiredo, I. Jesus, J. T. Machado, J. Ferreira, and J. M. De Carvalho, "Towards the development of intelligent transportation systems," in *Intelligent Transportation Systems*, vol. 88, pp. 1206-1211, August 2001.

[2] L. E. Y. Mimbela and L. A. Klein, "Summary of vehicle detection and surveillance technologies used in intelligent transportation systems." Technical Report, 2007.

[3] A. Puri, "A survey of unmanned aerial vehicles (UAV) for traffic surveillance," Technical Report, 2005.

[4] B. Coifman, M. McCord, M. Mishalani, and K. Redmill, "Surface transportation surveillance from unmanned aerial vehicles," in *Proc. of the 83rd Annual Meeting of the Transportation Research Board*, 2004.

[5] H. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi, "Efficient road detection and tracking for unmanned aerial vehicle," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, pp. 297-309, 2015.

[6] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition (CVPR)*, pp. 593-600, 1994.

[7] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision", in *Proc. 7th Int. Joint Conf. Artificial Intelligence*, pp.674-679, 1981

[8] J. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," Intel Corporation, vol. 5, pp. 1-10, 2001.

[9] X. Liu, Z. Peng, H. Hou, and L. Wang, "Simulation and Evaluation of Using Unmanned Aerial Vehicle to Detect Low-Volume Road Traffic Incident," in *Transportation Research Board 94th Annual Meeting*, 2015.

[10] T. Moranduzzo and F. Melgani, "Car speed estimation method for UAV images," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Quebec City, 2014, pp. 4942-4945.

[11] D. Shukla and E. Patel, "Speed determination of moving vehicles using Lucas-Kanade Algorithm," *International Journal of Computer Applications Technology and Research*, vol. 2, pp. 32-36, 2013.

[12] E. Patel and D. Shukla, "Comparison of Optical Flow Algorithms for Speed Determination of Moving Objects," *International Journal of Computer Applications*, vol. 63, pp. 32-37, 2013.

[13] Z. Xin, Y. Chang, L. Li, and G. Jia-ning, "Algorithm of Vehicle Speed Detection in Unmanned Aerial Vehicle Videos," in *Transportation Research Board 93rd Annual Meeting*, 2014.

[14] A. C. Shastry and R. A. Schowengerdt, Airborne video registration and traffic-flow parameter estimation, *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 4, pp. 391 -405, Dec. 2005.

[15] The OpenCV Library, Bradski, G., *Dr. Dobb's Journal of Software Tools*, 2000.