DETECTING AND COUNTING VEHICLES FROM SMALL LOW-COST UAV IMAGES

Penggen Cheng[1], **Guoqing Zhou**[1,2] and **Zezhong Zheng**[2,3]
[1]School of Geosciences and Geomatics, East China Institute of Technology
56 Xuefu Road, Fuzhou City, Jiangxi 344000, P. R. China
[2]Department of Civil Engineering and Technology, Old Dominion University, Norfolk, VA 23529, USA
Tel: (757) 683-6234; Fax: (757) 683-5655; E-mail: gzhou@odu.edu
[3]School of Civil Engineering, Southwest Jiaotong University, Chengdu, Sichuan, 610031, China

ABSTRACT

In recent years, many civil users have been interested in unmanned aerial vehicle (UAV) for traffic monitoring and traffic data collection because they have the ability to cover a large area, focus resources on the current problems, travel at higher speeds than ground vehicles, and are not restricted to traveling on the road network. This paper presents a method for detecting and counting vehicles from UAV video flow. The algorithm for vision-based detection and counting of vehicles in monocular image sequences for traffic scenes have been developed. In the algorithm, video frame-to-frame matching to track vehicle is one of important steps. Dynamic vehicles are identified using both background elimination and background registration techniques. The background elimination method uses concept of least squares to compare the accuracies of the current algorithm with the already existing algorithms. The background registration method uses background subtraction which improves the adaptive background mixture model and makes the system learn faster and more accurately, as well as adapt effectively to changing environments. In addition, because of high data sampling rates of video flow, resampling of video flow is also analyzed and discussed. The objective of this research is to monitor activities at traffic intersections for detecting congestions, and then predict the traffic flow.

INTRODUCTION

Traffic congestion has been a significantly challenging problem. It has widely been realized that increases of preliminary transportation infrastructure e.g., more pavements, and widened road, have not been able to relieve city congestion. As a result, many investigators have paid their attentions on intelligent transportation system (ITS), such as predict the traffic flow on the basis of monitoring the activities at traffic intersections for detecting congestions. To better understand traffic flow, an increasing reliance on traffic surveillance is in a need for better vehicle detection such at a wide-area. Traditionally the high costs detectors are mounted at the edge of pavement. In recent years, many civil users have been interested in unmanned aerial vehicle (UAV) for traffic monitoring and traffic data collection because they have the ability to cover a large area, focus resources on the current problems, travel at higher speeds than ground vehicles, and are not restricted to traveling on the road network. This paper presents a method for detecting and counting vehicles from UAV video flow. This is because the vehicle tracks, or trajectories, can be measured over a length of roadway, rather than at a single point, which used to be measured traditionally using detectors. Thus, it is possible to measure true density instead of simply recording detector occupancy. In fact, the traditional traffic parameters are more stable than corresponding measurements from point detectors when averaging trajectories over space and time from a UAV-based video flow. The additional information, such as lane changes of vehicles from UAV-based video, acceleration/deceleration patterns, could lead to improved incident detection. The trajectory data could also be used to automate previously labor intensive traffic studies, such as examining vehicle maneuvers in weaving sections or bottlenecks.

UAV PLATFORM AND DATA COLLECTION

Old Dominion University was recently awarded a grant for development and test of low-cost small civilian UAV system under contract of U.S. National Science Foundation. The goal of this project will be to investigate and develop techniques for generating high-resolution video flow from UAV video for fast-response to time-critical event, such as incident. In collaboration with the Air-O-Space International L.L.C. (AOSI), a small and lower-cost

UAV system was implemented to collect aerial video stream. This system, as illustrated in Figure 1, is actually divided into two parts: UAV platform and UAV ground control. As for deployment of UAV platform, video camera, GPS receiver and INS sensor are amounted on a small UAV aircraft as well as a Command Parser is installed to respond to ground control instructions. The UAV ground control mainly includes:

- Monitoring UAV flying status,
- Downloading video stream and GPS/INS navigation information,
- Planning UAV flight route, and
- Remotely control UAV flight route, height, and attitude.

On April 3, 2005, high-resolution video as well as telemetry data from on board GPS/INS orientation system is collected using the UAV system. The collected UAV video is a MPEG-encoded stream captured by a non-metric Topica Color CCD video camera (TP-6001A), whose video frame size is 760 × 480 pixels^2. The video flow remains basically at a nadir looking. The UTC time from onboard GPS were overlaid on to the video frames in the lower left hand corner. Telemetry data including UAV's position and attitude is recorded by GPS/INS sensors at 2-second interval. Beside video stream and telemetry data, ancillary data including USGS 10-meter grid DEM and 1-meter DOQQ referencing image covering experimental area are collected as well, which will provide ground control points for UAV system calibration and checking points for evaluating the accuracy of UAV data collection (see Table 1).

Table 1. Data collected in this project

Data	Functionality
Mepg-based Video stream	Collected by video camera onboard UAV and used to generate orthoimage
Telemetry Data	Collected by GPS/INS system onboard UAV and used to record UAV position and attitude
Ancillary data (USGS DEM/ Reference image)	Collected over the experimental area near AOSI office in Picayune, Mississippi and used for collecting ground control points (GCPs) to solve camera parameter or providing check points to evaluate accuracy of orthoimages.

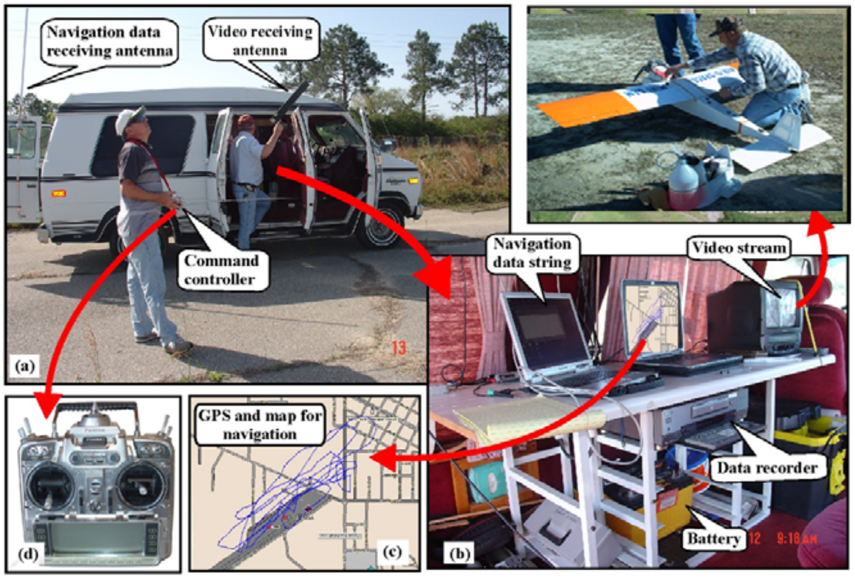


Figure 1. Small low-cost UAV developed by Zhou et al. (2006).

UAV-BASED VIDEO FLOW PROCESSING

Real-time UAV Video Flow Resampling

Because the original video is recorded at a sampling rate of 30 frame per second, a resampling has to be conducted in order to reduce the information redundant. Thus, we used Microsoft® DirectX set up architecture, called *DirectShow®*, for streaming media on the Microsoft Windows® platform. The core of DirectShow® is to use a modular architecture, called a *filter graph*, to mix and match different software components, called *filters*, for specific task related to multimedia streams. In this paper, a *filter graph (*Figure 2), is implemented for the purpose of real-timely re-sampling desired video frames from collected MPEG-encoded UAV video stream. Because each *Filter* in Figure 2 can respond to user's requirements and how stream data go through filters is fully controlled simply by issuing high-level command or requirement calls such as "Run" (to move data through the graph) or "Stop" (to stop the flow of data) or "sampling" (to extract any time-stamped sample from stream data), directly accessing MPEG-decoded stream and seeking any sample frame seamlessly in memory is achievable, which means video re-sampling can be achieved with high efficiency. (Please refer Microsoft® DirectX developing documents for more specific explanation on using Microsoft® DirectShow to process video stream.) As an example of a video pair of frame, $<N_i, N_{i+1}>$ (i >1), the resampling steps included:

1. Extracting frame N_{i+1} from video stream and calculate coarse parallax vector (dx_{i+1}, dy_{i+1}) for frame pair $<N_i, N_{i+1}>$. Suppose that sampling time interval between last pair of frame $<N_{i-1}, N_i>$ is T_i and its average parallax vector calculated from all tie points is (dx_i, dy_i), and a desired sampling time interval is T_{i+1} (e.g., one second), we can calculate time-stamp for frame N_{i+1} using $\sum_{k=1}^{i+1} T_k$ (the first frame N_0 is supposed to be re-sampled at beginning, i.e., 0 second). Through simple linear interpolation, coarse parallax vector for the a video pair $<N_i, N_{i+1}>$ is calculated by $(dx_{i+1}, dy_{i+1}) = (dx_i, dy_i)\times T_{i+1}/ T_i$;
2. Generating tie point candidates for the video pair $<N_i, N_{i+1}>$. With the coarse parallax vector (dx_{i+1}, dy_{i+1}), the potential conjugate points in the frame N_{i+1} can be predicted based on the feature points extracted from the frame N_i using the cross correlation algorithm of the feature points between the frame N_i and N_{i+1}, tie points are determined based on local extrema. To enlarge pull-in range, hierarchical matching based on pyramid-image is introduced in this step.
3. Detecting blunders of the tie point candidates and giving quantitative estimation T_e for the quality of rest tie points. With two-view co-planar geometry constrain (explained later) constructed by conjugated points and air line, the tie point candidates are refined by removing detected blunder and calculating statistic error T_e as quantitative estimation for the quality of rest tie points.
4. Repeating the steps 1 through 3 for generating valid tie points for video pair $<N_i, N_{i+1}>$ when T_e is bigger than a given threshold $T_{0.}$ When T_e is not bigger than a given threshold T_0, tie points for the next video pair $<N_{i+1}, N_{i+2}>$ will be generated by enlarging time-stamp for re-sampling the frame N_{i+2}.

The proposed re-sampling method in Step 4 is based on one fact that the liner interpolation for parallax vector can be guaranteed when the sampling time interval is very small although the UAV flies unsteadily. In other words, even though the UAV was flying at poor condition, the reliability of liner interpolation of the parallax vector still can be guaranteed.

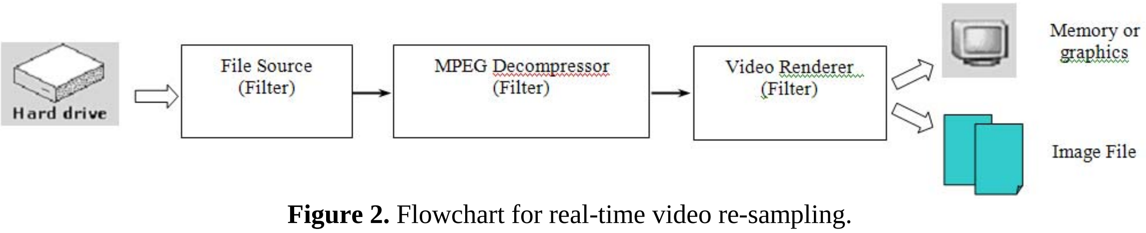


Figure 2. Flowchart for real-time video re-sampling.

UAV-based Vehicle Tracking

Many algorithms for detecting and counting vehicles from video have been developed in the computer vision community. A detailed review may be beyond of this paper. Briefly, the different tracking approaches for video data can be classified as follows (Koller, et al, 1993; Sullivan, 1992):

- **3D Model-Based Tracking:** 3D model-based vehicle tracking method is to use the prior 3D vehicle model to carry 3D-to-2D matching technology. The critical weakness of this approach lies on the detailed geometric object models. It is unrealistic to expect to be able to have detailed models for all vehicles that could be found on the roadway

- **Region-Based Tracking:** This approach is based on the traditional gray-based image matching technology to track vehicle over time. This approach works fairly well in free-flowing traffic. However, under congested traffic conditions, vehicles partially occlude one another instead of being spatially isolated, which makes the task of segmenting individual vehicles difficult. Moreover, this algorithm is very sensitive to the image quality.
- **Active Contour-Based Tracking:** The basic idea of this approach is to have a representation of the bounding contour of the object and keep dynamically updating it (Koller, et al.; 1994a; 1994b). The advantage of having a contour based representation instead of a region based representation is reduced computational complexity. However, this approach is also sensitive to occlusions and image quality.
- **Feature-Based Tracking:** This approach is tracking distinct feature, such as corner, edge line. The advantage of this approach is to tolerate partial occlusion, and not sensitive to image quality relative to other tracking methods.

This paper presents our algorithm for detecting and counting the vehicle from UAV-based video data. The main idea is to track the pixel from consecutive video frames, since the video flow can provide us with the overlapped image pair at any extent. The major steps include as follow.

A. Interesting Point (IP) Feature Extraction. Moravec operator (Moravec **) is implemented to extract Interested Point (IP) features from the left frame. A little but special improvement to the Moravec operator in this paper is that we cover target frame with certain virtual grid (e.g. each cell of this grid is 40×40 pixel) and only one point feature is extracted, if it exists, see Figure 3, to each cell respectively. With this improvement, not only time-cost for point feature extraction is reduced but also possible ambiguity problems for point matching are greatly relieved.

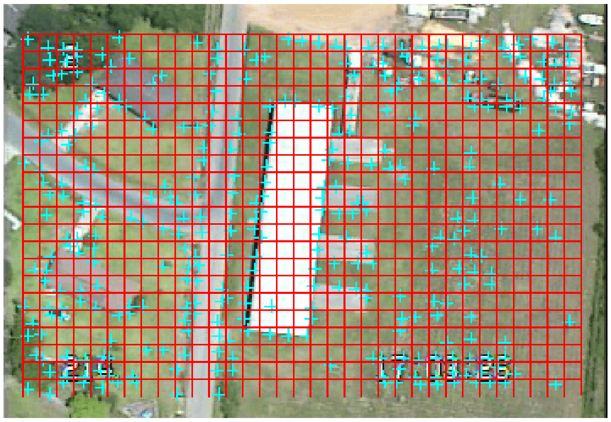


Figure 3. Method for extracting IP within grid and the perdition of the conjugate point (blue cross).

B. Conjugate Point Prediction. Based on hypothesized coarse parallax vector, corresponding conjugate points in the right frame can be predicted. Figure 3 shows the predicted conjugate points in the right frames.

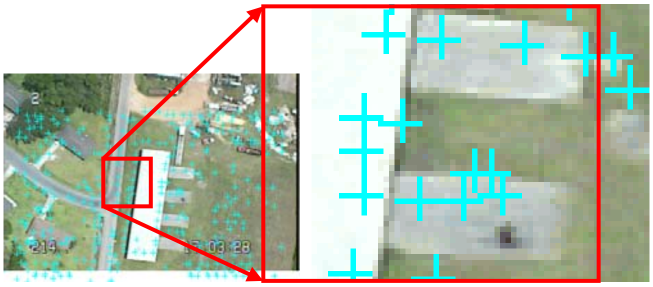


Figure 4. The perdition of the conjugate point.

C. Implementing Cross Correlation. For area-based matching, the similarity between gray value windows is usually defined as a function of the differences between the corresponding gray values. In this paper, this function is the cross correlation coefficient: $\rho = S_{xy} / \sqrt{S_{xx} S_{yy}}$. (where: Sxy is covariance function of "target window" and "matching window" ; Sxx, Syy is variance function of "target" or "matching" window respectively). By comparing calculated cross correlation coefficient to a given threshold, conjugate points for each extracted point feature can be determined. Figure 5 show the result when the cross correlation calculation is implemented to Figure 3 and Figure 4.

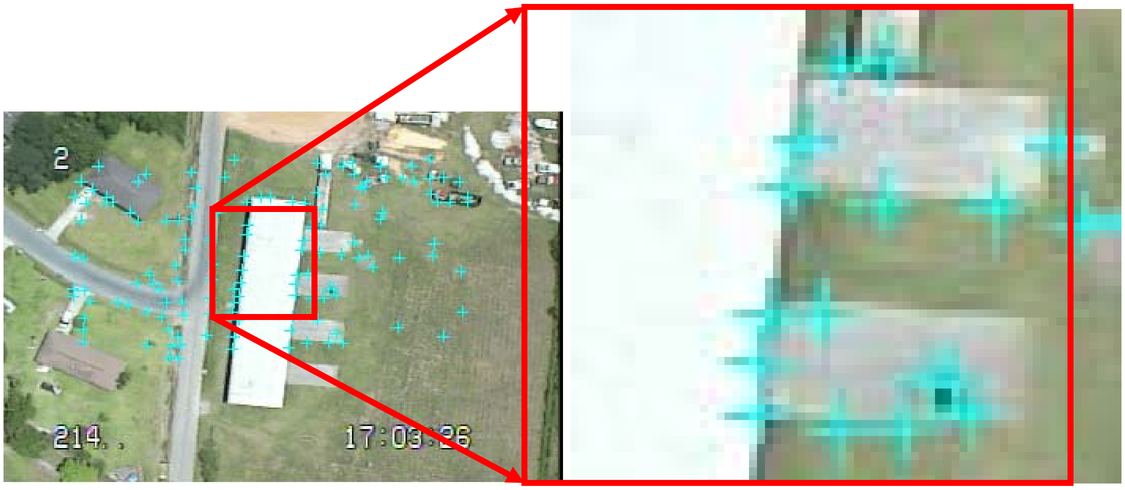


Figure 5. Tie point candidates from cross correlation.

D. Verification of Matched Results. Since the cross correlation method only applies the local extrema information to determine conjugate points, the reliability analysis for global consistency will be given to detect mis-matched conjugate points in last step and estimate the quality of tie point candidates.

Suppose that <a$_1$, a$_2$> be one conjugate point in tie point candidates, the homologous rays S$_1$a$_1$, S$_2$a$_2$ from project center S$_1$, S$_2$ respectively should intersect at the same space point A, namely, vector $\overrightarrow{S_1 a_1}, \overrightarrow{S_2 a_2}$ and $\overrightarrow{S_1 S_2}$ (called air base B) are co-planar and can be expressed by:

$$F = \overrightarrow{S_1 S_2} * (\overrightarrow{S_1 a_1} \times \overrightarrow{S_2 a_2}) = 0 \qquad (1)$$

Actually, from the view of relative orientation in photogrammetry community, Eq. 1 is the function of five relative orientation parameters of $\varphi, \ \kappa, \ \varphi', \ \omega', \ \kappa'$ and can be linearized as [14]:

$$F = F_0 + \frac{\partial F}{\partial \phi_1} \nabla \phi_1 + \frac{\partial F}{\partial \phi_2} \nabla \phi_2 + \frac{\partial F}{\partial \kappa_1} \nabla \kappa_1 + \frac{\partial F}{\partial \kappa_2} \nabla \kappa_2 + \frac{\partial F}{\partial \omega_2} \nabla \omega_2 = 0 \qquad (2)$$

for each conjugate point, one error equation can be set up based on Eq. 2 and thus, for all tie point candidates, all error equations can be expressed in the form of matrix, i.e.,

$$-v = AX - L \qquad (3)$$

With redundant observations, five relative orientation parameters *X* and standard deviation can be estimated. With the standard deviation σ_0 , tie point candidates with errors beyond $3\sigma_0$ can be thought of blunder and removed on the basis of probability theory. Figure 5 is the result when coplanar constrain is applied to Figure 6.

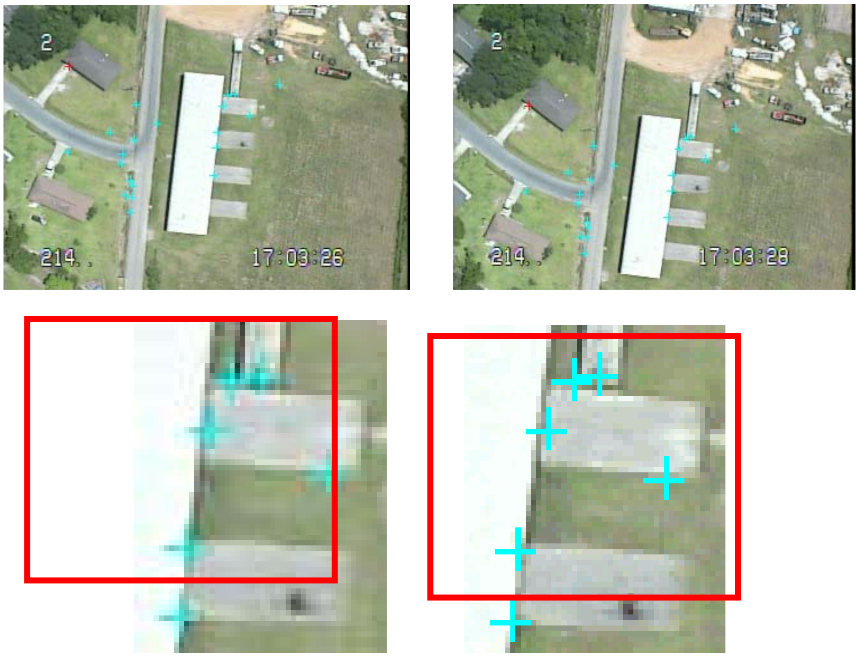


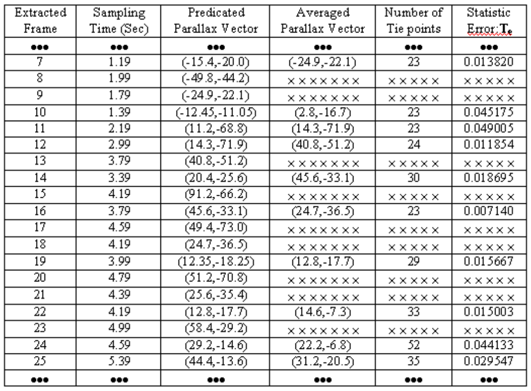
Figure 6. Verification of matched tie points.

EXPERIMENTS ON DETECTING AND COUNTING VEHICLES

Table 2 is an example of the results of UAV-based video match. As seen from Table 2, using average parallax vector calculated from all tie points in the last video stereo frames (e.g., averaged parallax vectors of 7 th frame and 10th frame are 2.8, - 16.7, respectively) as prediction for next consecutive video stereo frame (e.g. predicted parallax vectors of 10 th frame and 11th frame are 11.2, - 68.8, respectively), which is solved out by linearly interpolating, i.e., (11.2,-68.8) = (2.8,-16.7)×(2.19sec-1.39sec)/(1.39sec-1.19sec) is reasonable. The predicted parallax vector is appropriately close to average the parallax vector in the same stereo frame (e.g. averaged parallax vector of 10 th frame and 11th frame is (14.3, - 71.9)). Also, when the average parallax vector calculated from last stereo frame is not good as predicted one for next stereo frame (e.g. 14th and 15th frame), self-adjusting parallax vector by reducing time interval to generate new stereo frame (e.g. 14 th and 16th frame) does work well.

With the proposed matching method, the vehicle can be tracked. Figure 6 is the result of tracked vehicle. With the tracked vehicle, the accounting vehicle from UAV-based video can be obtained.

Table 2. Example of searching tie points (source: Wu and Zhou (2006))



Extracted Frame	Sampling Time (Sec)	Predicated Parallax Vector	Averaged Parallax Vector	Number of Tie points	Statistic Error:T_e
●●●	●●●	●●●	●●●	●●●	●●●
7	1.19	(-15.4,-20.0)	(-24.9,-22.1)	23	0.013820
8	1.99	(-49.8,-44.2)	×××××××	×××××	×××××
9	1.79	(-24.9,-22.1)	×××××××	×××××	×××××
10	1.39	(-12.45,-11.05)	(2.8,-16.7)	23	0.045175
11	2.19	(11.2,-68.8)	(14.3,-71.9)	23	0.049005
12	2.99	(14.3,-71.9)	(40.8,-51.2)	24	0.011854
13	3.79	(40.8,-51.2)	×××××××	×××××	×××××
14	3.39	(20.4,-25.6)	(45.6,-33.1)	30	0.018695
15	4.19	(91.2,-66.2)	×××××××	×××××	×××××
16	3.79	(45.6,-33.1)	(24.7,-36.5)	23	0.007140
17	4.59	(49.4,-73.0)	×××××××	×××××	×××××
18	4.19	(24.7,-36.5)	×××××××	×××××	×××××
19	3.99	(12.35,-18.25)	(12.8,-17.7)	29	0.015667
20	4.79	(51.2,-70.8)	×××××××	×××××	×××××
21	4.39	(25.6,-35.4)	×××××××	×××××	×××××
22	4.19	(12.8,-17.7)	(14.6,-7.3)	33	0.015003
23	4.99	(58.4,-29.2)	×××××××	×××××	×××××
24	4.59	(29.2,-14.6)	(22.2,-6.8)	52	0.044133
25	5.39	(44.4,-13.6)	(31.2,-20.5)	35	0.029547
●●●	●●●	●●●	●●●	●●●	●●●

Figure 7. Tracked and accounted vehicle from video.

CONCLUSIONS

This paper presents UAV-based real-time video data processing with emphasis on the video frame matching algorithm for vehicle detection, tracking and accounting. The algorithm takes advantage of the inherent characteristics of UAV video system of parallax, which is used to predict the corresponding points, and delete the mismatching points from video stream. This paper also discussed re-sampling of the video data. The experimental results demonstrated that the proposed image matching algorithm is efficient and effective. The vehicles from UAV can be detected, tracked and accounted.

ACKNOWLEDGMENTS

This paper was funded by U.S. National Science Foundation under the contract number of NSF 344521. Dr. Jun Wu, a postdoctoral researcher, conducted all code development and data processing. Authors thank him very much for his important contribution.

REFERENCES

Koller, D., K. Daniilidis, H. Nagel, 1993. Model-based object tracking in monocular image sequences of road traffic scenes, *International Journal of Computer Vision*, 10: 257-281.

Koller, D., J. Weber, J. Malik, 1994a. Robust multiple car tracking with occlusion reasoning, *ECCV*, Stockholm, Sweden, pp 189-196.

Koller, D., J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, S. Russell, 1994b. Towards robust automatic traffic scene analysis in real-time, *ICPR*, Israel, vol. 1, pp. 126-131.

Moraver, H.P., 1977. Towards automatic visual obstable avoidance, In *Proc. 5th Int. Joint Conf. Artificial Intell*, Cambridge, MA, p.584.

Sullivan, G., 1992. Visual interpretation of known objects in constrained scenes, *Phil. Trans. Roy. Soc (B)*, 337: 361-370.

Wu, Jun and G. Zhou, 2006. Low-cost unmanned aerial vehicle (UAV) video orthorectification, *2006 IEEE International Geoscience and Remote Sensing Symposium*, Denver, Colorado, USA, July 31, 2006 - August 4, 2006.

Zhou, G., J. Wu, S. Wright, and J. Gao, 2006. High-resolution UAV video data processing for forest fire surveillance, *Technical Report to National Science Foundation*, Old Dominion University, Norfolk, Virginia, USA, August, 82 p.