Vehicle tracking in wide area motion imagery from an airborne platform

Adam W.M. van Eekeren, Jasper R. van Huis, Pieter T. Eendebak, Jan Baan

TNO, Oude Waalsdorperweg 63, 2597 AK The Hague, The Netherlands

ABSTRACT

Airborne platforms, such as UAV's, with Wide Area Motion Imagery (WAMI) sensors can cover multiple square kilometers and produce large amounts of video data. Analyzing all data for information need purposes becomes increasingly labor-intensive for an image analyst. Furthermore, the capacity of the datalink in operational areas may be inadequate to transfer all data to the ground station. Automatic detection and tracking of people and vehicles enables to send only the most relevant footage to the ground station and assists the image analysts in effective data searches.
In this paper, we propose a method for detecting and tracking vehicles in high-resolution WAMI images from a moving airborne platform. For the vehicle detection we use a cascaded set of classifiers, using an Adaboost training algorithm on Haar features. This detector works on individual images and therefore does not depend on image motion stabilization. For the vehicle tracking we use a local template matching algorithm. This approach has two advantages. In the first place, it does not depend on image motion stabilization and it counters the inaccuracy of the GPS data that is embedded in the video data. In the second place, it can find matches when the vehicle detector would miss a certain detection. This results in long tracks even when the imagery is of low frame-rate. In order to minimize false detections, we also integrate height information from a 3D reconstruction that is created from the same images. By using the locations of buildings and roads, we are able to filter out false detections and increase the performance of the tracker. In this paper we show that the vehicle tracks can also be used to detect more complex events, such as traffic jams and fast moving vehicles. This enables the image analyst to do a faster and more effective search of the data.

Keywords: detection, tracking, airborne platforms, UAV, image processing, motion-in-motion, wide area motion imagery, 3D reconstruction.

1. INTRODUCTION

Because airborne platforms are recording large amounts of video data (especially with WAMI sensors), extracting the relevant events is a time-demanding task for image analysts. Given the fact that often only a fraction of the video footage contains events of interest[1] and that the capacity of the datalink in operational areas may be inadequate to transfer all data to the ground station in real-time, automatic detection and tracking of people, vehicles and buildings is useful in order to send only the most relevant footage to the ground station. This can be done in a real-time scenario (e.g. to follow a specific target), and in an offline scenario, where the information is analyzed after all data is stored in a database (e.g. to obtain patterns of life in the area of operation).
Depending on the resolution of the imagery, different kinds of information can be extracted from the obtained tracks, such as specific actions of persons[2–4], typical routes of vehicles and anomalies. To assist the analyst best, the tracks need to be long enough and of sufficient quality (preferably no track breaks and no track take-overs).
In literature a wide variety of trackers can be found. A recent survey is published by Smeulders et al.[5]. From this survey it becomes clear that most trackers are evaluated only on a small number of videos and that their performance is good in only a subset of conditions (illumination changes, clutter, etc.). In this respect the presented method in this paper is not different; it is evaluated on a limited dataset[6] and designed given the challenges of this dataset. The main challenges are that the data has a very high resolution (13.200 x 8.800 pixels), has a low framerate (2 fps) and is captured with four cameras from a moving aerial platform. The goal is to detect and track the vehicles in this dataset.
For the vehicle detection a cascaded set of classifiers is applied, using an Adaboost training algorithm on Haar features[7].

This detector works on individual images and therefore does not depend on image motion stabilization. Height information is used to remove false detections. Height information can be obtained from e.g. a public available database. In our experiments we used height information from a 3D reconstruction that was created from the WAMI data.
For the vehicle tracking we use a local template matching algorithm, based on the ideas presented in[8]. This approach has two advantages. In the first place it does not depend on image motion stabilization and it counters the inaccuracy of the GPS data that is embedded in the video data. In the second place it can find matches when the vehicle detector would miss a certain detection. This results in long tracks even when the vehicle displacement between the frames is pretty large due to the low frame-rate. Furthermore, it is shown that the resulting vehicle tracks can be used to detect more complex events, such as traffic jams and fast moving vehicles.
The setup of the paper is as follows. Section 2 describes the tracking method used. In Section 3 the camera system setup and the data are described. The results of the experiments are presented in Section 4. Finally, conclusions will be drawn in Section 5.

2. METHOD DESCRIPTION

In this section the approach is described which is used for tracking vehicles in WAMI images. A block diagram of this approach is depicted in Figure 1.

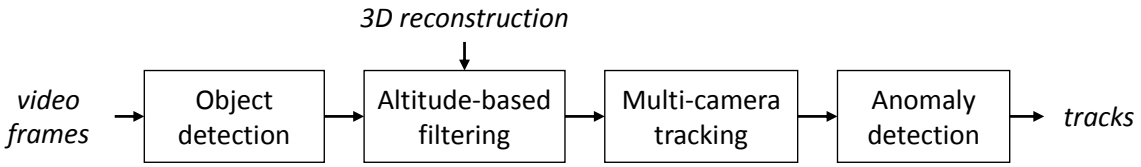


Figure 1. Flowchart of the method to detect and track vehicles in WAMI images and to find anomaly tracks.

2.1 Object detection

The first step is the detection of the vehicles. For this purpose a static object detector is used, which is trained using an Adaboost training algorithm on Haar features[7]. This detector works on individual images and therefore does not depend on image motion stabilization. Another advantage of a static object detector is that it is able to detect vehicles that are standing still (e.g. parked or waiting at traffic lights), something which would not be possible with background subtraction techniques.
The training set is constructed by annotating the vehicles in only six video frames. Due to the high resolution of the imagery, these few frames contain several hundred vehicles. All training objects are then rotated by a multitude of 22.5 degrees, in order to obtain "horizontal" training images with an aspect ratio of 25 x 15. On this set of images the vehicle detector is trained.
This detector is then applied on all image frames in the video. Each frame is rotated from -90 to 90 degrees with in-between steps of 22.5 degrees. This way the "horizontal" vehicle detector is able to detect vehicles heading in all directions. A certain vehicle may be detected in different rotations of the same video frame. From the resulting overlapping detections, only the detection with the highest confidence is maintained.

Figure 2. Example of a vehicle that is detected in 3 different rotations. Only the detection with the highest confidence is maintained (the one in the middle).

2.2 Height-based detection filtering

The detections obtained in the first step are filtered based on their altitude. In order to determine these altitudes, first a 3-dimensional reconstruction of the area was created[9]. Using this 3D reconstruction, a height map for each video frame could be made. These height maps were then used to estimate the height of each detection, see the examples in Figure 3. This filtering could remove a number of false vehicle detections, although also vehicles in car parks located on top of buildings were wrongly discarded. Note that due to missing data in the height maps it was not possible to estimate the height of all detections. These detections (marked in blue) were not removed from the original set.

Figure 3. Two frames showing detected vehicles where the color indicates the estimated height of the vehicle. Green: < 5m, yellow: > 5m & < 10m, red: > 10m, blue: no height could be estimated.

2.3 Multi-camera tracking

The main part of the presented approach consists of multi-camera tracking, which is depicted in the block diagram in Figure 4. At each point in time the steps in the block diagram are processed.

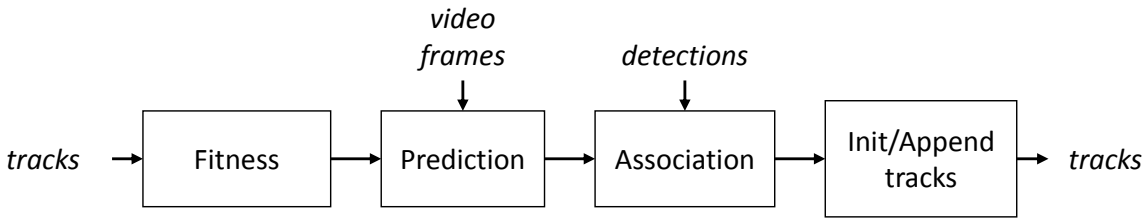


Figure 4. Flowchart of multi-camera tracking.

Fitness

The first step in the tracking is to determine the fitness of all the tracks. A track is *killed* (meaning that it will not be updated anymore) under one of the following conditions: 1) no detection is associated to the track within the last 10 seconds, 2) the time that the last association is made is larger than the duration of the track.

Prediction

A track prediction is the expected position of an active vehicle track, based on the previous detections of that vehicle. The prediction of the tracks is performed by template matching. This approach has the advantage that 1) it counters the inaccuracy of the GPS data that is embedded in the video data and 2) it can find matches when the vehicle detector would miss a certain detection. This results in long tracks even when the imagery has a low frame-rate.
First a new location of a track is predicted in world coordinates (latitude, longitude). This predicted position is then converted to image coordinates of the camera which has the predicted position in its field-of-view. The calculated image coordinates are the center of the search area for the template matching, which is done in the Fourier domain by calculating the sum of squared differences. For speed considerations the search area is typical 50 pixels in each direction outside the template.

The template matching returns the 10 best matches. If the cost of the best match (C_1) is significantly smaller than the cost of the second best match (C_2); $C_2 > 1.2*C_1$, then the best match becomes the new prediction. Otherwise the center of the search area will be the new prediction.

Association
The detections are associated with the tracks based on overlap of rotated bounding boxes. If a rotated bounding box of a detection overlaps more than 50% with the predicted rotated bounding box of the track, the detection is added to the track.

Init/Append tracks
In this step the non-associated tracks and detections are treated. First, all detections that were not associated to a track become new tracks. Second, all tracks that got no detection associated, are appended with a prediction only.

2.4 Event detection

Event detection algorithms are designed to retrieve specific events or anomalies that might be of interest for an operator, i.e. to convert the thousands of tracks into user-relevant information. Examples of information that can be extracted from the data are:

1. Road density maps or speed maps;
2. High-speed vehicles (with respect to the speed map);
3. Road blocks / traffic jams;
4. Vehicles following each other.

3. EXPERIMENTS

3.1 System description

The experiments are performed on WAMI images obtained with a CorvusEye™ 1500 system[6]. This system is equipped with four high resolution cameras with a resolution of 6.600 x 4.400 pixels each, running at 2 frames per second. The resolution of the overall stitched image is 13.200 x 8.800 pixels (= 116 Mpix). An example of the camera setup is depicted in Figure 5 from which it is clear that the field-of-view of the cameras slightly overlap.

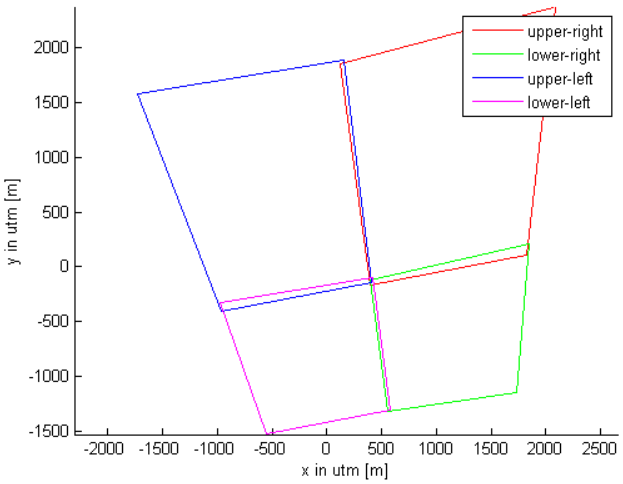


Figure 5. Field-of-views of the four cameras plotted on the ground plane in shifted UTM coordinates.

3.2 Data description

For the tracking 3156 WAMI images are used, which show the downtown of Rochester, NY, USA. The images are captured from an airplane which flew one circle in a timespan of approximately 6,5 minutes. An example of a stitched image is depicted in Figure 6.

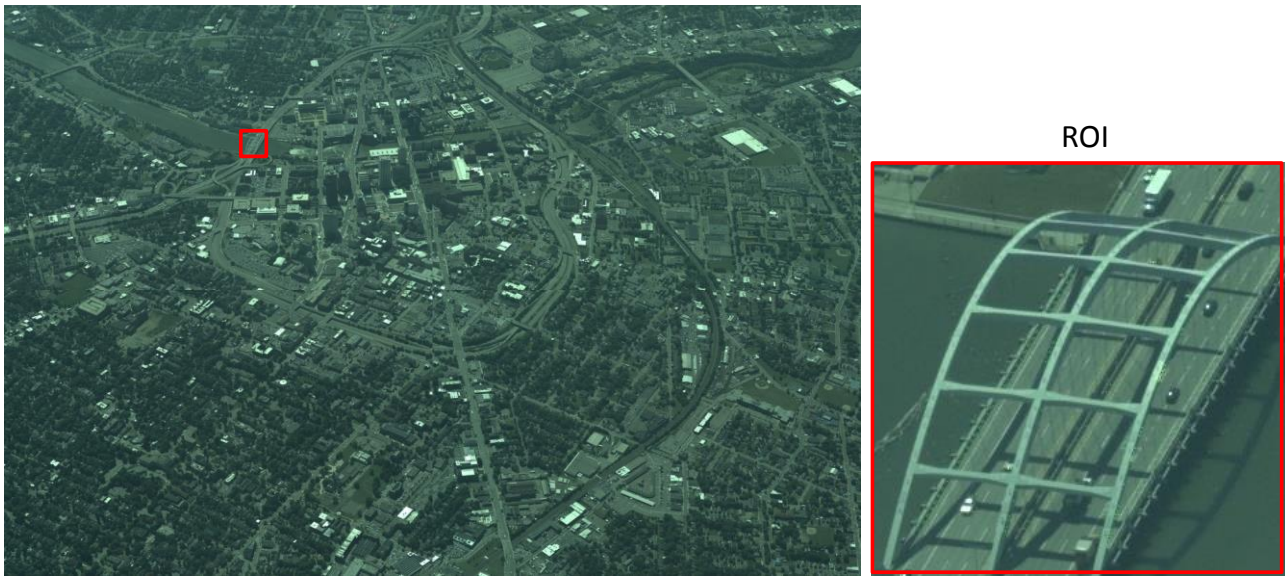


Figure 6. Left: example of a stitched image showing downtown Rochester. Right: zoom-in showing the amount of detail which is present.

The ground resolution varies between 0,18 meter/pixel to 0,31 meter/pixel.

4. RESULTS

4.1 Tracking

After tracking, all tracks are linearly interpolated at the time instances: 1) at which no detections were added to the tracks and 2) which lie between time instances containing associated detections. In Figure 7 a histogram is plotted of all tracks with a track length >= 9 frames. This set contains 40.905 tracks and 2.629.545 detections, resulting in a mean track length of 64,3 frames (= 32 seconds). These results allow to perform event and anomaly detection (see section 4.2).

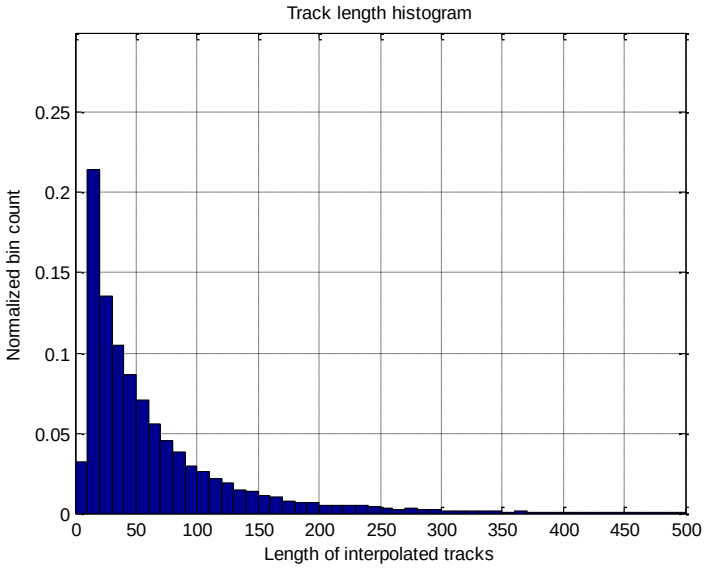


Figure 7. Histogram of track length of tracks with length >= 9 frames.

To get an idea of where the tracks lie on the ground plane, a plot is made of all latitude-longitude coordinates of the tracks in the above described set, but with the extra constraint that a track has a minimum length of 200 meter. The plot of the resulting 5093 tracks is depicted in Figure 8.

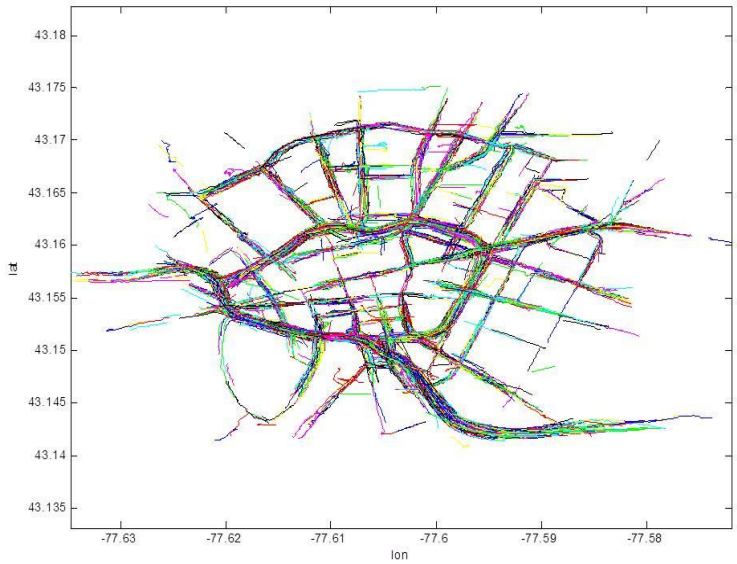


Figure 8. Tracks longer than 200 meter plotted on the ground plane in latitude-longitude coordinates.

An example of a specific track is depicted with some screenshots in Figure 9.

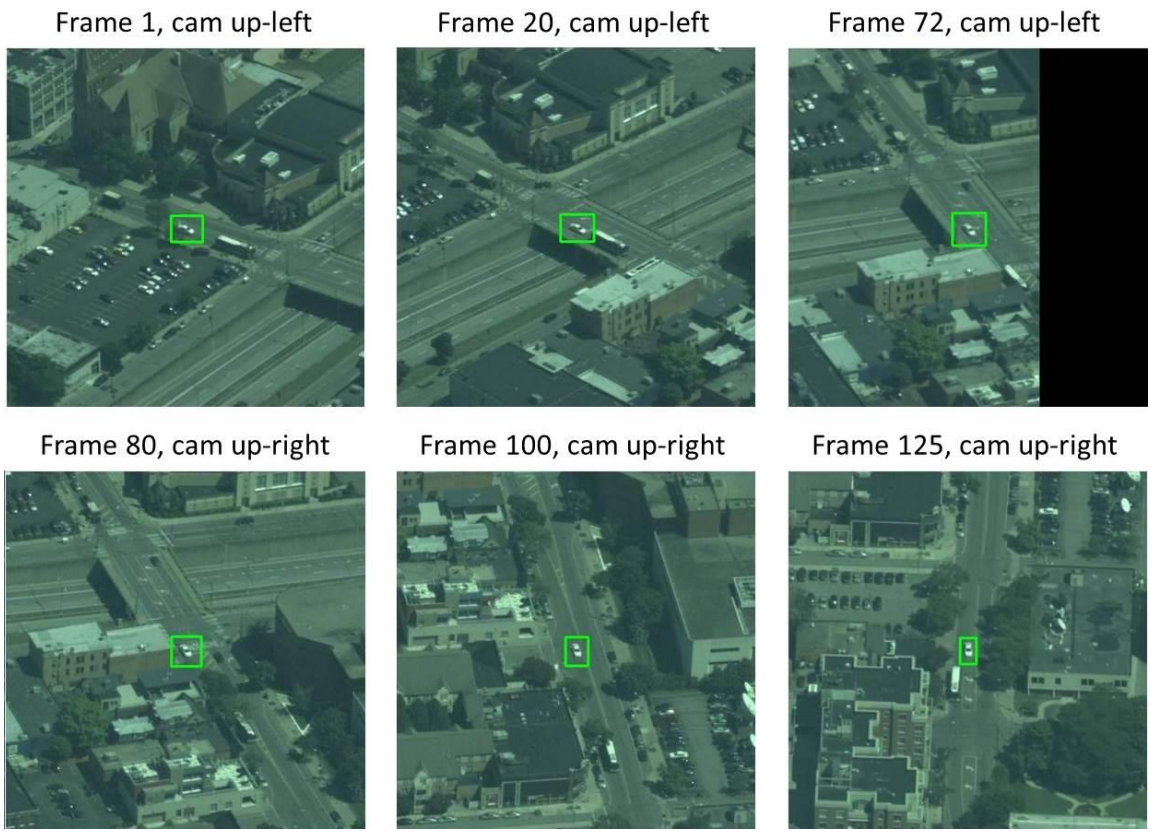
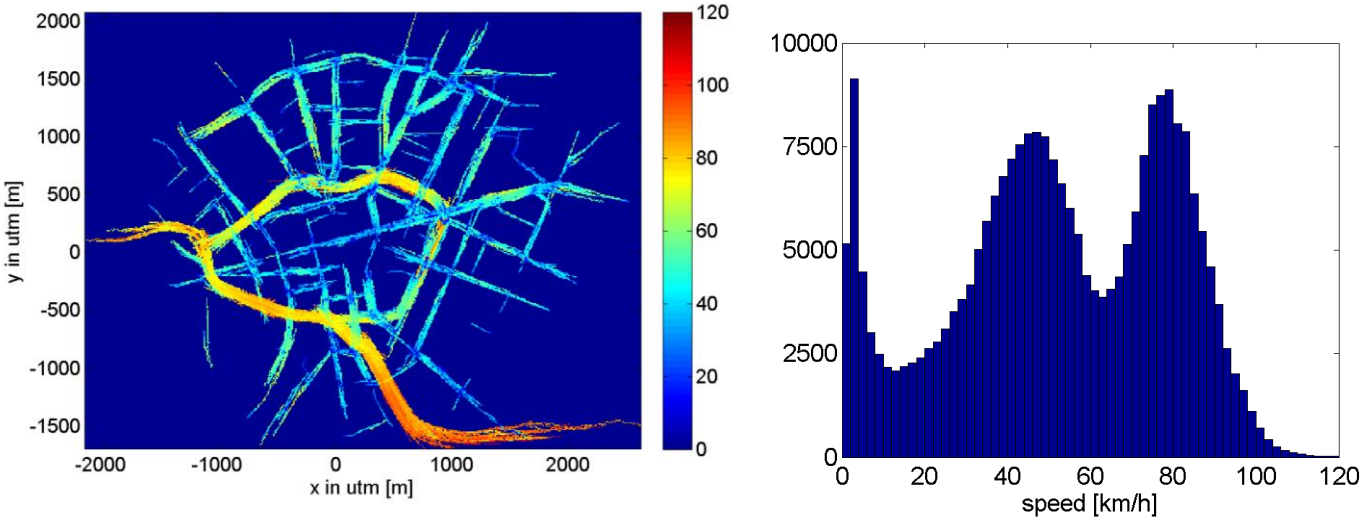


Figure 9. Screenshots of a moving vehicle, which is tracked over two cameras (up-left and up-right).

4.2 Event and anomaly detection

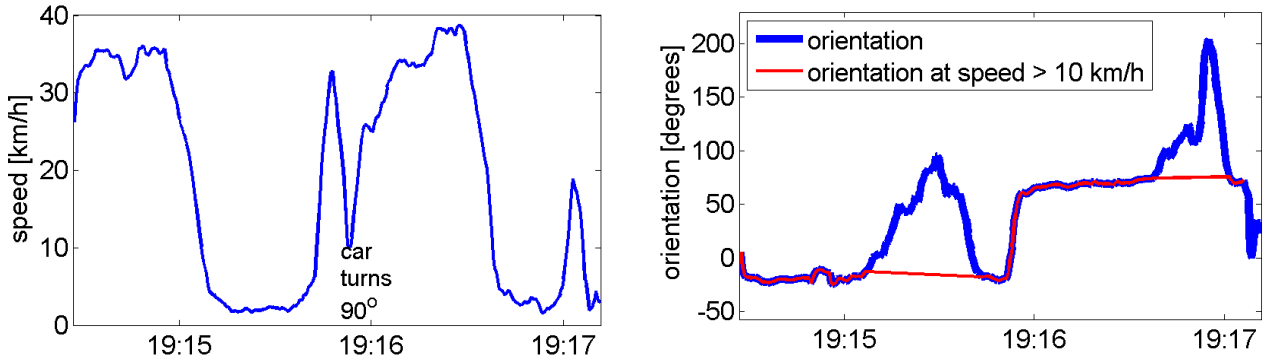
Given the tracks all kind of information can be extracted. One of them is a speed map. In Figure 10a the speed of each track in km/h is plotted on the ground plane. From this plot it becomes clear where the main roads are located and what the speed limitation is on the main road near the city center (80 km/h).



(a) (b)

Figure 10. Left: speed (km/h) of all tracks plotted as a heat map on the ground plane in shifted UTM coordinates. Right: Histogram of speed for all tracks.

Another interesting plot is depicted in Figure 10b where the speed distribution of all tracks is shown. In this plot three peaks can be distinguished: 1) tracks having no speed (parked cars), 2) tracks with speed ~45 km/h (cars driving at secondary roads and 3) tracks with speed ~80 km/h (cars driving at main roads).



(a) (b)

Figure 11. Left: speed of a specific track. Right: orientation of the same track. From both plots it can be deducted that around time 19:16 the car makes a turn of approximately 90 degrees.

Events such as turning of a car can also be detected easily from all the track information. An example is depicted in Figure 11, where on the left hand side the sharp dip in the speed curve indicates that the car slows down before the turn and accelerate afterwards. On the right hand side the orientation shows that the car has made a turn of approximately 90 degrees. Note that the red line shows only the orientation at speed larger than 10 km/h, because at low speeds the orientation estimation is inaccurate.

4.3 Processing performance

This research did not include the optimization of the processing speed of the algorithms. Currently the total processing speed is around one minute per frame on a standard PC using Matlab. An optimized implementation on a dedicated system could result in (near) real-time performance. This would enable the system to be used for on-board event and anomaly detection, which can support real-time tactical operations.

5. CONCLUSIONS

The results show that the presented method is capable of tracking vehicles in challenging multi-camera WAMI data. Processing 6,5 minutes of video at 2 fps results in 40.905 tracks of at least two seconds (including 2.629.545 detections). This gives an overall mean track length of 64,3 frames (= 32 seconds). It is shown that with these tracking results additional information can be extracted such as the speed of each track segment, which gives an operator insight in the localization of the main and secondary roads. But also anomalies, such as a vehicle driving through a 'green wave', can be easily extracted from the tracking results.
Automatic processing of large amounts of full motion video can be useful in both real-time and offline scenarios. In a real-time scenario the data connection may have insufficient bandwidth to transfer all video data to the ground station. Onboard tracking and analysis of moving objects can be used to determine the relevant data to send to the ground station. In an offline scenario the automatic processing can be used to support the operator or image analyst. He can use the information to effectively interpret large amounts of data or to perform data searches faster or more effectively. The possibility to specify thresholds for anomalies is a very powerful tool. For real-time tactical operations it is very important to get an automatic alert if any vehicle approaches a certain area of interest from a suspicious direction and with an above average speed.

ACKNOWLEDGEMENTS

The authors would like to thank Harris Corporation for the availability of the CorvusEye™ 1500 imagery[6]. This research is done within the Unmanned Systems program (V1340) in which TNO researches the use of different unmanned systems.

REFERENCES

[1] Trinh, H., Li, J., Miyazawa, S., Moreno, J.., Pankanti, S., "Efficient UAV video event summarization," 21st Int. Conf. Pattern Recognit. ICPR, 2226–2229, IEEE (2012).
[2] Bouma, H., Hanckmann, P., Marck, J.-W., Penning, L., Hollander, R. den., Hove, J.-M. ten., van den Broek, S., Schutte, K.., Burghouts, G., "Automatic human action recognition in a scene from visual inputs," SPIE Def. Secur. Sens., 83880L – 83880L, International Society for Optics and Photonics (2012).
[3] Burghouts, G., Schutte, K., Bouma, H.., Hollander, R. den., "Selection of negative samples and two-stage combination of multiple features for action detection in thousands of videos," Mach. Vis. Appl. **25**(1), 85–98 (2014).
[4] van Eekeren, A. W. M., Dijk, J.., Burghouts, G., "Detection and tracking of humans from an airborne platform," SPIE Secur. Def., 92490S – 92490S – 7, SPIE (2014).
[5] Smeulders, A. W. M., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A.., Shah, M., "Visual Tracking: an Experimental Survey," IEEE Trans Pattern Anal Mach Intell (2014).
[6] CorvusEye™ 1500 imagery courtesy of Harris Corporation.
[7] Viola, P.., Jones, M., "Rapid object detection using a boosted cascade of simple features," Proc. 2001 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2001 CVPR 2001 **1**, 511–518 (2001).
[8] Lewis, J. P., "Fast template matching," Vis. Interface **95**, 15–19 (1995).
[9] Furukawa, Y.., Ponce, J., "Accurate, Dense, and Robust Multiview Stereopsis," IEEE Trans. Pattern Anal. Mach. Intell. **32**(8), 1362–1376 (2010).