FAST VEHICLE DETECTION AND TRACKING IN AERIAL IMAGE BURSTS

Karsten Kozempel and Ralf Reulke

German Aerospace Center (DLR e.V.), Institute for Transportation Systems
Rutherfordstraße 2
12489 Berlin
karsten.kozempel@dlr.de, ralf.reulke@dlr.de

KEY WORDS: aerial, image, detection, tracking, matching

ABSTRACT:

Caused by the rising interest in traffic surveillance for simulations and decision management many publications concentrate on automatic vehicle detection or tracking. Quantities and velocities of different car classes form the data basis for almost every traffic model. Especially during mass events or disasters a wide-area traffic monitoring on demand is needed which can only be provided by airborne systems. This means a massive amount of image information to be handled. In this paper we present a combination of vehicle detection and tracking which is adapted to the special restrictions given on image size and flow but nevertheless yields reliable information about the traffic situation.
Combining a set of modified edge filters it is possible to detect cars of different sizes and orientations with minimum computing effort, if some a priori information about the street network is used. The found vehicles are tracked between two consecutive images by an algorithm using Singular Value Decomposition. Concerning their distance and correlation the features are assigned pairwise with respect to their global positioning among each other. Choosing only the best correlating assignments it is possible to compute reliable values for the average velocities.

1 INTRODUCTION

1.1 Motivation

The gathering of traffic information is a base for all kinds of traffic modeling, simulation and prediction for tasks like emission reduction, efficient use of infrastructure or extension planing of the road network as well as the intervention and resource planing. Next to the use of inductive loops, Video Image Detection Systems (VIDS) have become a common alternative due to their low price as well as their simplicity and effort of installation. Furthermore inductive loops can't cover the whole road network and a lot of data has to be estimated. Especially during mass events or disasters with huge congestions or road blocks, they can't yield reliable information.
For this special purpose the German Aerospace Center (DLR e.V.) developed the ANTAR system for airborne traffic monitoring on demand. During the soccer world cup 2006 it was successfully applied to gather traffic data and predict traffic situation in three German cities (Ruhé et al., 2007). Based on this the DLR is developing the ARGOS system for wide-area traffic monitoring (fig.1). It contains next to a radar system the 3K-Cam, a device of three digital cameras with 16 mega pixels each. Together they cover an area of 2,5 km x 0,7 km with a resolution of 20 cm at an altitude of 1000 m over ground. Additionally a GPS/IMU-unit is used to record positioning and orientation data for every image taken. Thereby the achieved image data gets orthorectified and georeferenced on-board which means that the images arriving the traffic detecting software can be used as map images with given orientation and scale. A fact that makes measuring distances and computing velocities less complex.
In the first chapter the conditions related to the observation system are explained as well as the published work on this area. The second chapter describes the used algorithms, a modified edge filter for fast vehicle detection and an extended singular value decomposition concerning distances and correlations for tracking in very short sequences. After this the results with a few examples are presented. Finally a conclusion with considering possible further research will close the paper.

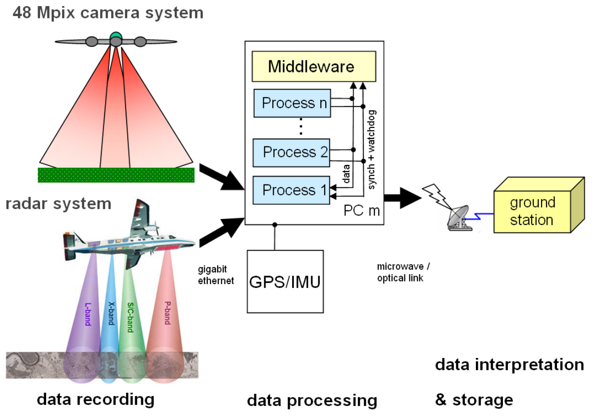

Figure 1: Traffic monitoring system ARGOS

1.2 Special conditions

There are two special points to consider while developing detection and tracking. It should be respected that the preprocessed images depending on their altitude over ground can be very large, in the shown case 25-30 mega pixels. That's why the detecting algorithm should be rather fast than exact. Already the previous system ANTAR demonstrated that for an overview of the traffic situation a completeness of two thirds is acceptable.
Due to the mentioned size of the images (original size is 16 mega pixels) they cannot be transmitted continuously. After a burst of a few images (2-4) the stream is cut to save them. Therefore it is not necessary to implement a complex tracking filter which needs a long period to adapt to the scene.

1.3 Related work

A grand variety of approaches in vehicle detection as well as in object tracking has been released in the last years.
Detection methods can be divided into two groups, depending on the kind of model being used. The use of explicit models

for example is explained in (Haag and Nagel, 1999), (Moon et al., 2002), (Hinz, 2004) and (Ernst et al., 2005). In (Haag and Nagel, 1999) a very extensive database of about 400 different three-dimensional car models is used to predict the appearance of vehicles including their shadow cast. In (Hinz, 2004) the author uses not only the shadow but additionally the luminance and reflectivity of the car's surface as well which of course is more expensive to process. Next to shape and shadow in (Zhao and Nevatia, 2003) they try to recognize the windshield of vehicles. The final decision is made by a Bayesian Network. Most of them have a very reliable detection rate of more than 90 percent but a long computing time. In the papers (Moon et al., 2002) and (Ernst et al., 2005) they use rather simple two-dimensional models for detection. While in (Ernst et al., 2005) the authors search in the edge filtered image for rectangular objects of certain size in (Moon et al., 2002) they already shape the edge filter to a rectangle of expected car size. Both of them provide a fast and acceptable detection rate using additional information about street area and direction.

The use of implicit models is explained in (Grabner et al., 2008) and (Lei et al., 2008). In (Grabner et al., 2008) the author supposes to use a learning AdaBoost algorithm which is robust and fast by making a lot of cascaded weak decisions. In (Lei et al., 2008) they train a support vector machine with the SIFT descriptors of selected cars and non-cars. But both of these approaches have to be trained with lots of positive and negative samples before working independently. Additionally it is not easy to cover all cases of illumination and environment. That's why many leaning algorithms have to be trained for every situation separately. Another easy approach for detection of moving cars without using any model is explained in (Reinartz et al., 2006) where they detect all moving objects in adjacent images by computing the normalized difference image. But as the georegistration of the images often is less exact than the pixel size, the images have to be coregistered first. On the other hand only moving objects can be detected while traffic jams or queues in front of a traffic light would be ignored.

Concerning tracking there are lots of publications using optical flow and Kalman or particle filters to predict the expected displacement and appearance in following images. (Haag and Nagel, 1999) and (Nejadasl et al., 2006) pursued this approach which is not easy to realize in the special case of only two or three adjacent images. In (Lenhart and Hinz, 2006) they use especially triplets of images to determine the best match between at least three states which can be described as a kind of prediction. Another good idea for the special case of very short bursts is presented in (Scott and Longuet-Higgins, 1991) and improved in (Pilu, 1997). The authors use singular value decomposition of a distance matrix to match a group of features to another one with respect to the relative positions of all features among each other. (Pilu, 1997) later extends the approach by adding the correlation between pairs of features.

2 APPROACH

2.1 Preprocessing

To identify the active regions as well as the orientation of images among each other they have to be georeferenced, which means their absolute geographic position and dimension have to be defined. Related to the GPS/IMU information and a digital terrain model the image data gets projected into GeoTIFF images, which are plane and oriented into north direction. This is useful to combine the recorded images with existing datasets like maps or street data. To avoid examining the whole image data, only the street area given by a database is considered.

2.2 Detection

For providing fast detection of traffic objects in the large images a set of modified edge filters, that represent a two-dimensional car model, is used. Recent tests showed that the car's color information does not yield better results in detection than its gray value. Therefore the original images are converted into gray images. This conversion saves two thirds of filtering time. As there is additional information about street area and orientation this knowledge is used as well. The databases provided by Navteq (www.navteq.com) and Atkis (www.atkis.de) for example contain that information about the street network. For every street segment covered by the image a bounding box around it is cut out. The subimage is masked with the street segment to only use the filters on traffic area. We use neither a Hough transformation for finding straight edges nor a filter in shape of the whole car, as mentioned in (Moon et al., 2002). But we create four special shaped edge filters to represent all edges of the car model, which are elongated to the average expected size and turned into the direction given by the street database (fig.2 and 3). To avoid filtering for all different car sizes, we only shift the filter answers (fig.4) to the expected car edges within a certain range. This has the same effect as positioning the filter kernels around an anchor point. In the conjunction image of the four thresholded and shifted edge images remain blobs at the position, where all four filters have answered strong enough to the related edge filter. The regions remaining (fig.5) become thinned by a non-maxima

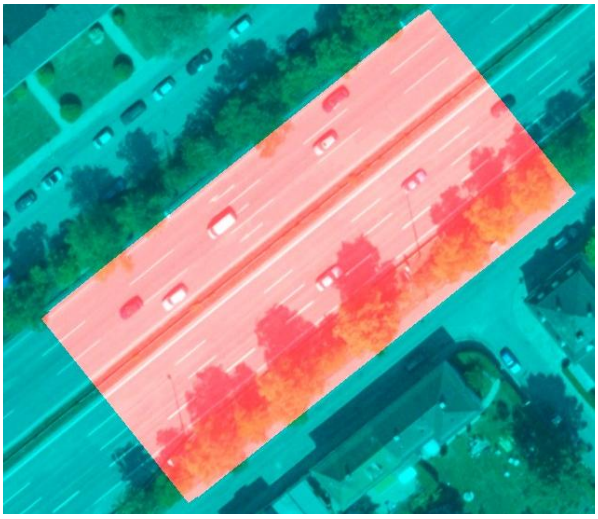

Figure 2: Mask based on Navteq street segments

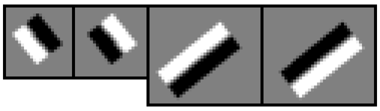

Figure 3: The associated filter kernels

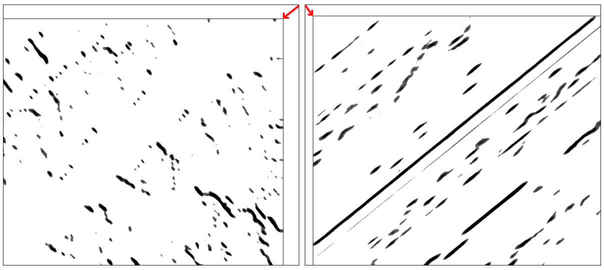

Figure 4: The shifted and thresholded filter answers 2 and 3

suppression until one pixel each is left representing the car's center. Fig.6 shows the regions left related to the cars that caused them.

For bigger vehicles like trucks the same filter answers are used. To recognize long edges without using new filters, the given answers of the side edges are shifted along the side of the car and always conjuncted with each other.

To avoid cars being detected twice, all observations are tested pairwise for their distances among each other. Some observations have more than one maximum, or vehicles are detected twice between two neighboring street segments. With respect to their size and orientation, objects below a certain distance to each other are discarded while only the one with the strongest intensity remains.

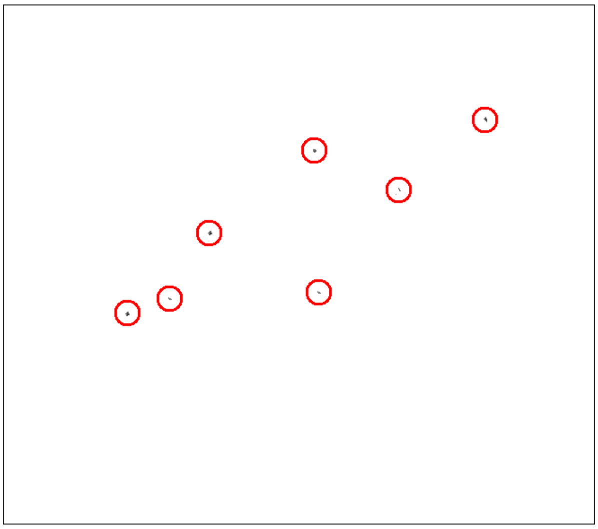

Figure 5: Regions where all filters answered

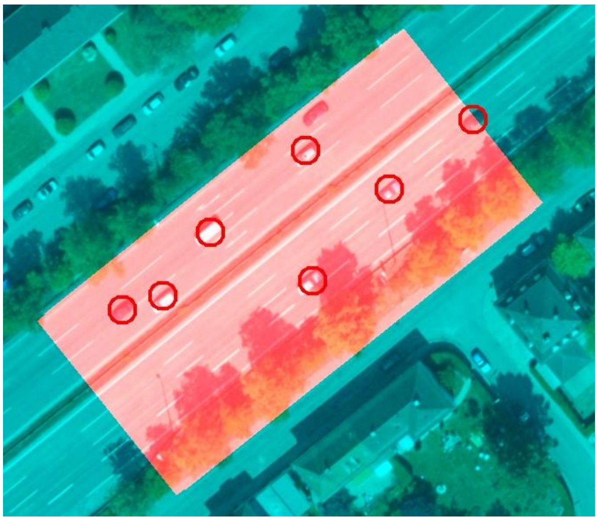

Figure 6: The detected cars

2.3 Tracking

As there are only short bursts of images, a classic Kalman filter cannot really be used. As already mentioned Lenhart's approach in (Lenhart and Hinz, 2006) uses prediction for image triplets. This works just in case there are triplets. Bursts with less than three images, which appear as well, have to be handled different. That's why we only consider relations between two consecutive images. Scott and Longuet-Higgins suggest in (Scott and Longuet-Higgins, 1991) a singular value decomposition as a kind of one-to-one correspondence with respect to the positions of all neighboring objects. This is more an association than a real tracking as only the last image's information is used. If I and J are two images with m features I_i and n features J_j we build a proximity matrix $\mathbf{G}$ with the Gaussian-weighted distances G_{ij} between every feature I_i and J_j.

$$G_{ij} = e^{-r_{ij}^2/2\sigma^2} \qquad (1)$$

where $r_{ij} = ||I_i - J_j||$ is is the euclidean distance. So the elements G_{ij} decrease monotonically with the distance. The parameter σ defines the degree of interaction between the features. A small value enforces local and a big one rather global interaction. It is recommended to choose σ as large as the average expected distance the feature pairs have.

The next step is to perform a singular value decomposition of the proximity matrix $\mathbf{G}$. The Algorithm is provided by a lot of software libraries. Here the one in OpenCV was used.

$$\mathbf{G} = \mathbf{TDU}^T \qquad (2)$$

After the SVD the matrices $\mathbf{T}$ and $\mathbf{U}$ are orthonormal matrices and the diagonal matrix $\mathbf{D}_{m \times n}$ contains the positive singular values as diagonal elements in descending order. As the third and last step a new matrix $\mathbf{P}$ has to be computed by

$$\mathbf{P} = \mathbf{TEU}^T \qquad (3)$$

where $\mathbf{E}$ is the changed diagonal matrix $\mathbf{D}$ with all elements replaced by $\mathbf{1}$. The resulting matrix $\mathbf{P}$ has the same dimensions as $\mathbf{D}$ but by the algorithm the values P_{ij} for good pairings have been amplified while those for bad ones have been reduced. So if P_{ij} is the greatest element in column and row the two features I_i and J_j are in a 1:1 correspondence with one another.

Furthermore Pilu (Pilu, 1997) extends the algorithm for feature-based stereo matching by using the cross correlation of two features next to their distance. So the SVD-association can be used for images concerning the similarity of a certain window around their features. Adding this (Gaussian-weighted) information to the proximity matrix $\mathbf{G}$ the elements G_{ij} result as follows:

$$G_{ij} = e^{-(C_{ij}-1)^2/2\gamma^2} \cdot e^{-r_{ij}^2/2\sigma^2} \qquad (4)$$

where the left term is the Gaussian-weighted function of the normalized correlation coefficient C_{ij} between the features I_i and J_j. The parameter γ determines how fast the values decrease with C_{ij}. During our tests the best values lie between 0.4 and 1.0.

3 RESULTS AND DISCUSSION

3.1 Detection

The computing time and the accuracy of detection always depend on the number, size and quality of street segments given by the database. In the first example (shown in fig.6) only a broad highway in Munich has been tested without any smaller streets being considered. The processing of the 28 mega pixels large image took 30 seconds (Athlon 64 X2, 2.2 GHz, 2 GB RAM). The 96 vehicles were counted manually as ground truth and compared with the detected vehicles. The varying detection rates caused by varying thresholds are shown as the red graph in fig.7 and 8. As one can see there is always a trade-off between completeness and correctness. The more sensitive the thresholds are set the more false positives they will find. The graph shows the detection rate (number of true detected cars/real number of cars) in relation to the rate of false positives (number of false detected objects/number of detected objects). To be honest the false positives rate is not very objective, as the number of false detected objects does not depend on the real number of cars, and could turn out very bad just in case there is only one car in the image. Therefore in (Lei et al., 2008) they consider the FP-number in relation to the length of streets. A still better way would be to take the street area, for example 'false positives per hectare'. In this example there is an optimal point, where detection reaches 80 percent while the FP-rate is only ten percent or one car per hectare.

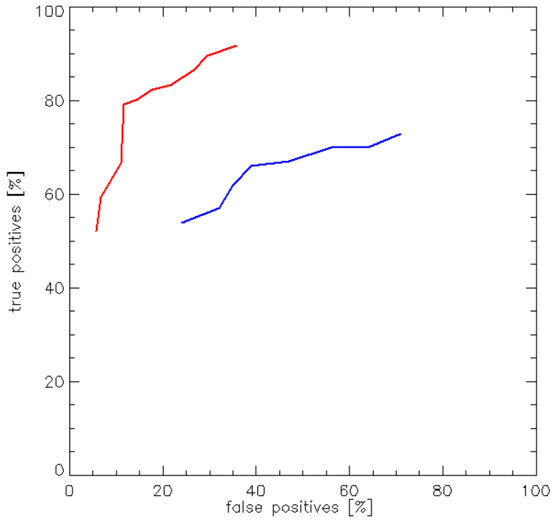

Figure 7: Detection rates on a highway (red) and narrow streets (blue) depending on false positives per detected cars

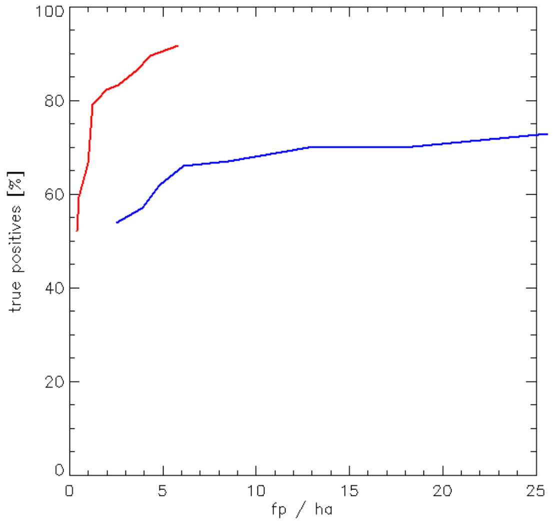

Figure 8: Detection rates on a highway (red) and narrow streets (blue) depending on false positives per area

A rather bad sample (the worst in our evaluation) represents the blue graph in fig.7 and 8 where more than 300 cars have been clicked by hand. If we consider streets of all sizes in the Munich suburban area, on the one hand the detection time takes longer (more than 60 seconds) and the results become worse as well. The detection rate stays around two thirds while only the number of false positives rises from 5 up to 25 per hectare.

A reason for the bad detection rate in the second example is the accuracy of street coordinates. As many smaller street elements are drawn next to the real street (fig.9) the algorithm misses many cars while detecting some rectangular structures next to the street. An approach to avoid this might be to improve the street accuracy by alternative street databases or street detection which should not be considered in this paper.

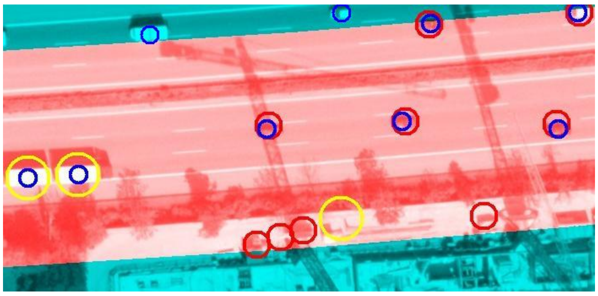

Figure 9: False positives and negatives due to incorrect coordinates (blue - existing car, red - found car, yellow - found truck)

3.2 Tracking

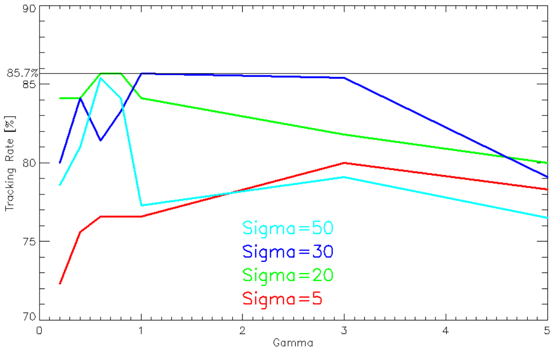

Figure 10: Correctness rate of tracks depending on the parameters σ and γ

We implemented the tracking algorithm as explained above by using the vehicles distances on UTM-projection and the normed correlation coefficient of all three color channels in a 20-by-20-pixels window around them. As the images cover an area of 700 on 1000 meters with hundreds of cars each, it is not easy to show how the whole set of tracks looks. That's why only one street was picked out for visualization. In fig.10 the resulting tracking rates depending on the parameters σ and γ are shown. As one can see the best results we get if σ is between 20 and 30. If the value is too small ($\sigma = 5$) the dependence of the positions among each other is not respected enough. This results not only in incorrect assigned pairs but also in crude mistakes by assigning objects together which are located very far from each other. This can strongly falsify the measured velocities. Furthermore γ should neither be too small nor too high. The best results yield values between 0.4 and 1.0. Around these settings a correctness of more than 80 percent (best value 85.7%) is achieved.

As for the average velocities it is rather important to accept correct tracks than getting all vehicles tracked, after the SVD the acceptance is bound to the correlation coefficient of a pairing. If the pairing next to its ranking in row and column does not pass a threshold for the CC, it is discarded although it might be correct. In fig.11 the remaining tracks are shown. In the upper half of the image 49 objects have been detected. 39 of them have been detected in the lower half as well which means they are possible to track. 36 of the objects have been assigned to another one, 30 of them were assigned correctly. After the thresholding with a CC of 0.9 still 26 of the 36 tracks remain. So from end to end

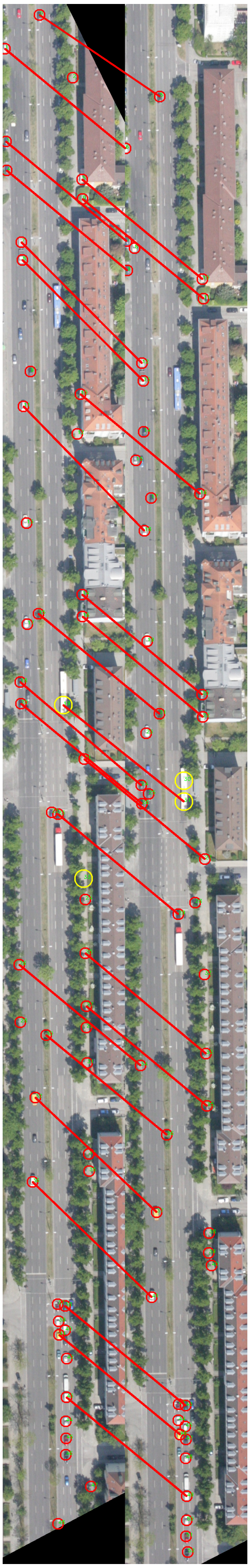

Figure 11: Tracked objects filtered by CC-threshold 0.9 (100% correct)

only 53 percent (26 out of 49) of all detected objects are found again and tracked, but with a correctness of 100 percent. Surely there should be more tests with more representative numbers, but we did not have enough reliable reference data yet. This will be done in the near future.

4 CONCLUSIONS AND FUTURE WORK

In this paper we presented a car detecting and a tracking algorithm which have been especially chosen and adapted to the given situation, the flying traffic monitor ARGOS. It was shown how they work and that they brought satisfying results depending on the environmental conditions. Furthermore it was shown, where the approach has problems and continuous work can be done.

Surely the system can be improved in some points and a few of them should be given here. First of all the street accuracy problem which could be easily solved by using another database. And it should be mentioned that there was already the attempt to use the more accurate street database Atkis. On the one hand the coordinates were indeed more exact and yielded slightly better detection rates, but on the other hand the database divides the street network into too small segments, which take a lot more time to process one by one. Additionally the achieved data should be mapped on Navteq segments, which would not be easy. So the next step is to integrate the newest version of the Navteq database being bought at the time.

Furthermore the edge detection could be optimized for example by running it on the GPU, but it has not been considered so far. Another idea is to compute the filtering in the frequency space. The Fourier-transformed images and filters just have to be multiplied in frequency space and transformed back. The only problem is that the filters change with every street segment, so there are four filters and four filtered images to be transformed every time. The approach was already explored, but the Fourier-transformation implemented in OpenCV needs longer than direct convolution, because it uses floating point numbers.

Next to this the detected cars could be verified by a more expensive algorithm like a Bayesian Network or a Support Vector Machine because some of the false positives do not look like a car at all. So they would be easy to discard.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Franz Kurz and Dr. Dominik Rosenbaum (Remote Sensing Technology Institute, German Aerospace Center, Oberpfaffenhofen) for providing the GeoTIFF images and navigation data.

REFERENCES

Ernst, I., Hetscher, M., Thiessenhusen, K., Ruhé, M., Börner, A. and Zuev, S., 2005. New approaches for real time traffic acquisition with airbone systems. Int. Archives of Photogrammetry, Remote Science and Spatial Information Sciences 36, pp. 68–73.

Grabner, H., Nguyen, T. T., Gruber, B. and Bischof, H., 2008. On-line boosting-based car detection fron aerial images. ISPRS Journal of Photogrammetry and Remote Sensing.

Haag, M. and Nagel, H., 1999. Combination of edge element and optical flow estimates for 3d-model-based vehicle tracking in traffic image sequences. International Journal of Computer Vision 35, pp. 295–319.

Hinz, S., 2004. Detection of vehicles and vehicle queues in high resolution aerial images. Photogrammetrie - Fernerkundung - Geoinformation (PFG) 3/04, pp. 201 – 213.

Lei, Z., Li, D. and Fang, T., 2008. Vehicle detection in high-resolution satellite imagery using sift features and support vector machines. ISPRS Journal of Photogrammetry and Remote Sensing.

Lenhart, D. and Hinz, S., 2006. Automatic vehicle tracking in low frame rate aerial image sequences.

Moon, H., Chellappa, R. and Rosenfeld, A., 2002. Performance analysis of a simple vehicle detection algorithm. Image and Vision Computing 20, pp. 1–13.

Nejadasl, F. K., Gorte, B. G. H. and Hoogendoorn, S. P., 2006. Optical flow based vehicle tracking strengthened by statistical decisions. ISPRS Journal of Photogrammetry and Remote Sensing 61, pp. 149–158.

Pilu, M., 1997. Uncalibrated stereo correspondence by singular value decomposition. Technical report, Digital Media Department, HP Laboratories Bristol.

Reinartz, P., Lachaise, M., Schmeer, E., Krauss, T. and Runge, H., 2006. Traffic monitoring with serial images from airborne cameras. ISPRS Journal of Photogrammetry and Remote Sensing 61, pp. 149–158.

Ruhé, M., Kühne, R., Ernst, I., Zuev, S. and Hipp, E., 2007. Airborne systems and data fusion for traffic surveillance and forecast for the soccer world cup. Technical report, German Aerospace Center.

Scott, G. L. and Longuet-Higgins, H. C., 1991. An algorithm for associating the features of two images. Proc. Royal Society London 244, pp. 21–26.

Zhao, T. and Nevatia, R., 2003. Car detection in low resolution aerial images. Image and Vision Computing 21, pp. 693–703.