AUTOMATIC VEHICLE TRACKING IN LOW FRAME RATE AERIAL IMAGE SEQUENCES

D. Lenhart, S. Hinz
Remote Sensing Technology, Technische Universitaet Muenchen, 80290 Muenchen, Germany
{Dominik.Lenhart|Stefan.Hinz}@bv.tu-muenchen.de

KEY WORDS: Vehicle Tracking, Traffic Monitoring, Traffic Parameters

ABSTRACT:

Traffic monitoring requires mobile and flexible systems that are able to extract densely sampled spatial and temporal traffic data in large areas in near-real time. Video-based systems mounted on aerial platforms meet these requirements, however, at the expense of a limited field of view. To overcome this limitation of video cameras, we developed a concept for automatic derivation of traffic flow data which is designed for commercial medium format cameras with a resolution of 25-40 cm and a rather low frame rate of only 1-3 Hz. Since resolution and frame rate are the most limiting factors, the focus of the first implementations and evaluations lies on the approach for automatic tracking of vehicles in image sequences of such type in near real-time. The tracking procedure relies on two basic components: a simple motion model to predict possible locations of previously detected vehicles in the succeeding images and an adaptive shape-based matching algorithm in order to match, i.e. recognize, the detected vehicles in the other images. To incorporate internal evaluations and consistency checks on which the decision of a correct track can be based, the matching is done over image triplets. The evaluation of the results shows the applicability and the potentials of this approach.

1. INTRODUCTION

1.1 Traffic Monitoring

Traffic monitoring is a very important task in today's traffic control and flow management. The acquisition of traffic data in almost real-time is essential to swiftly react to current situations. Stationary data collectors such as induction loops and video cameras mounted on bridges or traffic lights are matured methods. However, they only deliver local data and are not able to observe the global traffic situation. Space borne sensors do cover very large areas. Because of their relatively short acquisition time and their long revisit period, such systems contribute to the periodic collection of statistical traffic data to validate and improve certain traffic models. However, often, monitoring on demand is necessary. Especially for major public events, mobile and flexible systems are desired, which are able to gather data about traffic density, average velocity, and traffic flow, in particular, origin-destination flow. Systems based medium or large format cameras mounted on airborne platforms meet the demands of flexibility and mobility. While they have the capability of covering large areas, they can deliver both temporally and spatially densely sampled data. Yet, in contrast to video cameras, approaches relying on these types of cameras have to cope with a much lower frame rate.

A more extensive overview on the potential of airborne vehicle monitoring systems is given in (Stilla et al., 2004), while the use of aerial image sequences to derive traffic dynamics is studied in (Toth et al., 2003). There, it is also shown that the knowledge about traffic income and outgo directions allows a more precise and effective handling of traffic flow management.

1.2 Related Work

In the last decades, a variety of approaches for automatic tracking and velocity calculation have been developed. Starting with the pioneering work of Nagel and co-workers based on optical flow (Dreschler and Nagel 1982; Haag and Nagel, 1999), the usage of stationary cameras for traffic applications has been thoroughly studied. Further examples for this category of approaches are (Dubuisson-Jolly et al., 1996; Tan et al., 1998, Rajgopalan et al., 1999; Meffert et al., 2005). Some of the ideas incorporated in these approaches have influenced our work. Though, a straigtforward adoption is hardly possible since these approaches exploit oblique views on vehicles as well as a higher frame rate – both, however, at the expense of a limited field-of-view. Another group of approaches uses images taken by a photogrammetric camera with a high resolution of 5-15cm on ground (e.g., (Hinz, 2004)). Also, these approaches are hardly applicable since the vehicle's substructures which are necessary for matching a wire-frame model are no more dominant in images of lower resolution.

In (Ernst et al., 2005), a matured monitoring system for real time traffic data acquisition is presented. Here, a camera system consisting of an infrared and an optical sensor is mounted on slowly moving air vehicles like an airship or a helicopter, but also tests with aircrafts have been conducted. Traffic parameter estimation is based on vehicle tracking in consecutive image frames collected with a frame rate of 5 Hz and more. While the results are promising, a major limitation of this system is the narrow field of view (the width of one single road) due to the low flying altitude that is necessary to obtain a reasonable resolution on ground.

Considering the data characteristics, the most related approaches are (Reinartz et al. 2005) and (Lachaise, 2005). Like us, they use aerial image sequences taken with a frame rate of 1-3 Hz and having a resolution of 25-40cm. Vehicle detection is done by analyzing difference images of two consecutive frames. This method is quite robust to detect moving objects and to quickly find possible locations for car tracking. Yet, with this approach, it is not possible to detect cars that are not moving, which often also happens for active vehicles if they are stuck in a traffic jam or waiting at a traffic light or stop sign. Furthermore, tracking of detected vehicles includes an interactive component at the current state of implementation.

The boundary conditions of our work are primarily defined by the use of medium format cameras of moderate cost. They allow a large coverage and still yield a resolution of roughly

25cm. However, due to the high amount of data for each image, the frame rate must be kept rather low, i.e. 1 up to a maximum of 3 Hz. In the following, we will outline a concept to automatically detect and track vehicles which is designed to deal with these constraints. The main contribution presented here relates to the tracking procedure rather than the detection of the vehicles. We focused on this point first since the low frame rate is the most influencing factor of the overall concept, and the benefits and limitations of this module should be clearly analyzed. In addition, also some first results of automatic detection will be given.

2. OVERALL CONCEPT

The underlying goal of the concept outlined in the following is the fulfillment of near real time requirements for vehicle tracking and derivation of traffic parameters from image sequences. The general work flow is depicted in Fig. 1.

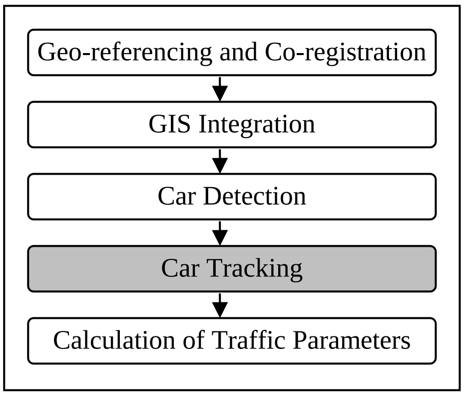

Figure 1. Work flow of online vehicle tracking

The images are co-registered and approximately geo-referenced after acquisition. This process is commonly supported by simultaneously recorded navigation data of an INS-/GPS-System. GIS road data, e.g. stemming from NAVTEQ or ATKIS data bases, are mapped onto the geo-referenced images and approximate regions of interest (RoI) are delineated(so-called road sections). Thus, the search area for the following automatic vehicle detection can be significantly reduced. For further processing, it is helpful to extract the road as well as their lanes in addition, since geo-referencing might not be accurate enough and GIS data rarely includes the position of individual lanes. An example for the automatic determination of lane sections using a slightly modified version of the road extraction system of Hinz & Baumgartner (2003) is shown in Fig 2. This example is generated by a stand-alone module and not yet incorporated into the automatic processing chain.

A car detection algorithm is supposed to deliver positions and, optionally, additional attributes such as boundary and direction constrained to the lanes within the RoI. Tests with matching wire frame models of cars showed only limited success due to the moderate ground resolution of 25-40cm. More promising results were obtained by a differential geometric blob detection algorithm similar to (Hinz, 2005), which has to be trimmed for colored blobs yet. Results of blob detection are shown in Fig. 3.

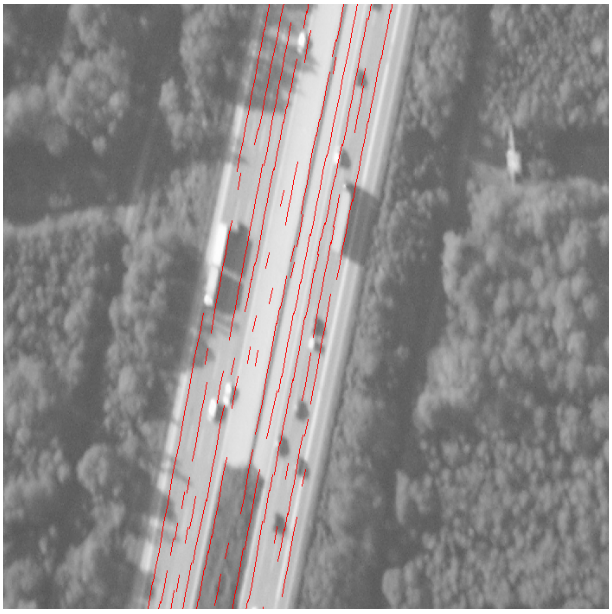

Figure 2. Intermediate result of lane extraction

Figure 3. Results of a blob detection

After their detection in the first image, the cars are tracked by matching them within the next two images. To this end, an adaptive shape-based matching algorithm is employed including internal evaluation and consistency checks (see details in Sect. 3). From the results of car tracking, various traffic parameters are calculated. These are most importantly vehicle speed, vehicle density per road segment, as well as traffic flow, i.e. the product of traffic density and average speed, eventually yielding the number of cars passing a point in a certain time interval.

In our tests of vehicle tracking, the first three parts are simulated, thereby accounting for potential impreciseness and uncertainty of the data. Their implementation is due to future work: *i*) The co-registration between image pairs is done by an affine 2D-transformation using least-squares optimization. This approximation seems reasonable, since our focus is on roads, which are generally almost planar objects. *ii*) GIS data have been simulated by digitizing road lines for each carriage way of a road. These lines will be referred to as "road polygons" in the sequel. They consist of "polygon points" *P*, while two of these enclose a "polygon segment" *L*. For each segment, the length as well as the orientation angle *ang(L)* are determined. *iii*) Cars are selected manually by digitizing the approximate center of the car including the shadow region since the shadow is an important indicator in detection and tracking a vehicle. However, since this step will be replaced by an automatic procedure in the near future (see Fig. 3), we will call them "detected vehicles" or "detected cars" in the following.

3. VEHICLE TRACKING

Before outlining algorithmic details of the tracking procedure in Sect. 3.2., we will first sketch the underlying vehicle motion model.

3.1 Vehicle Motion Model

The frame rate of the image sequences dictates the change of locations of a car, i.e. the possible maneuvers a car has undergone in the inter-frame time interval. Cars possibly move sideways and forward quite far within a period of half a second or more. Therefore, a motion model for predicting a vehicle's position in the next image is necessary.

3.1.1 Motion Model for Single Vehicles: We suppose that cars generally move in a controlled way, i.e. certain criteria describing speed, motion direction and acceleration should be met. To better incorporate the continuity of motion direction, we consider also a third image of the sequence. Figure 4 illustrates some of these cases. For instance, there should be no abrupt change of direction and change of speed, i.e. abnormal acceleration, from one image to the others. In general, the correlation length of motion continuity is modelled depending on the respective speed of a car, i.e., for fast cars, the motion is expected to be straighter and almost parallel to the road axis. Slow cars may move forward between two consecutive images but cannot move perpendicular to the road axis or backwards in the next image. These model criteria are incorporated in our tracking evaluation described in section 3.2.

3.1.2 Motion Model for Vehicle queues: In more complex traffic situations, the motion model can be extended to consider also vehicle queues. For images taken with a frame rate of 1-3 Hz, the car topology within a queue changes very rarely from one image to the other, although one could think of more complex queue motion models that describe the interaction of cars in a Markov-Chain manner.

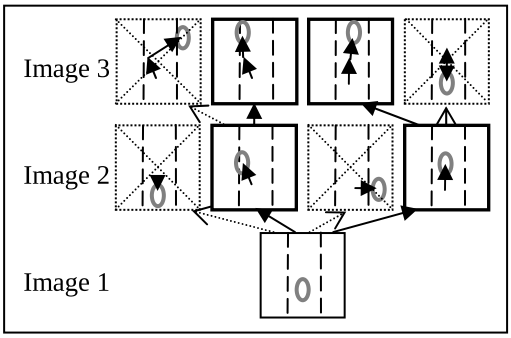

Figure 4. Examples for possible and impossible car movement

Hence, we currently analyze only pairs of cars as shown in Fig. 5. The distance of two cars following each other might increase or decrease, of course with a lower bound depending on the vehicles' speed. The trailing car may start to pass the leading car and change lanes. However, the cars cannot switch their relative positions.

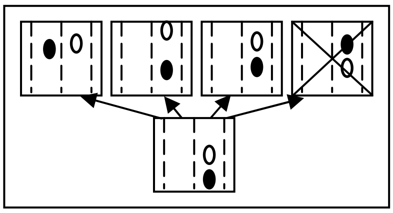

Figure 5. Vehicle queue behavior

3.2 Tracking procedure

In the current implementation, we focus on single car tracking in three consecutive images. Figure 6 shows the workflow of our tracking algorithm. As it can be seen, image triplets are used in order to gain a certain redundancy allowing an internal evaluation of the results. Of course, one could use more than three images for tracking. However, vehicles that move towards the flying direction only appear in few images so that the algorithm should also deliver reliable results for a low number of frames.

We start with the co-registration of the three images I1, I2, and I3, followed by car detection in I1 and the determination of a number of vehicle parameters which describe the actual state of a car, i.e. the distance to the road side polygon and the approximate motion direction (Sect. 3.2.1). Then, we create a vehicle image model C_1 by selecting a rectangle around the car. By using a shape-based matching algorithm, we try to find the car in the other images. In order to reduce the search, we select a RoI for the matching procedure based on the motion model (Sect. 3.2.2). The matching procedure delivers matches M_{12} in image I2 and the matches M_{13} in image I3. It should be mentioned, that both M_{12} and M_{13} contain multiple match results also including some wrong matches (see Fig. 7). As output of the matching algorithm, we receive the position of the match center.

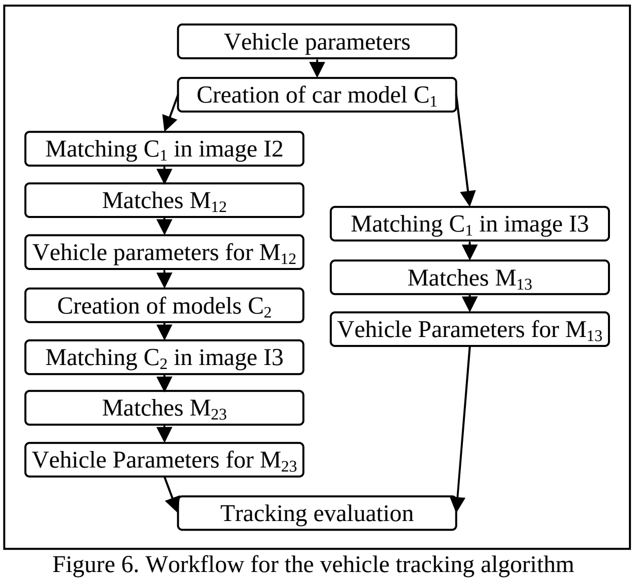

Figure 6. Workflow for the vehicle tracking algorithm

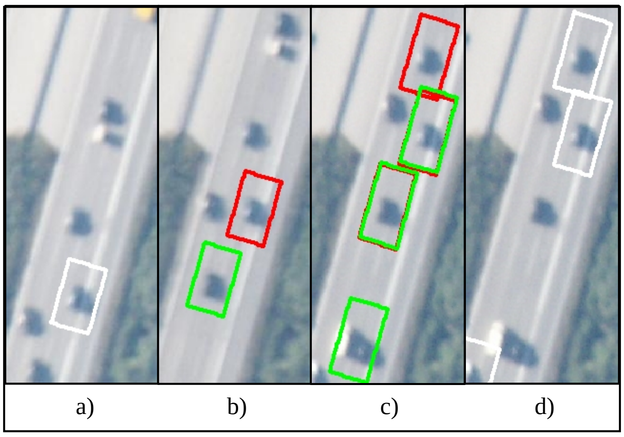

Figure 7. a) First image with detected car; b) second image with two matches M_{12} for C_1; c) third image with three matches M_{23} for each C_2 (corresponding matches are indicated by the same color; note the overlapping rectangles); d) third image with matches M_{13}

For each match M_{12}, vehicle parameters are calculated and new vehicle image models are created based on the match positions of M_{12}. These models are searched in image I3, eventually resulting in matches M_{23}, for which vehicle parameters are determined again. Finally, the results are evaluated and checked for consistency to determine the correct track combination of the matches (see Sect. 3.2.3).

3.2.1 Vehicle Parameters: The vehicle parameters are defined and determined as follows:

Distance to road polygon: The road polygon closest to a given vehicle is searched, and root point P_F is determined. This point is needed to approximate the direction of the car's motion.

Direction: A given vehicle's motion direction *dir(Car)* is approximated as a weighted direction derived from the three adjacent polygon segments, thus also considering curved road segments. The situation is illustrated in Fig. 8. The distances d_0 and d_1 between P_F and the end points of the central line segment L_n are determined. The weight of the angle of L_n is set to 1. The weight of the adjacent line segments' angles is computed using the relative distances d_0 and d_1. Note that d_0 is used to determine the weight of $ang(L_{n+1})$ while d_1 contributes to the weight of $ang(L_{n-1})$. This results in a higher weight for the angle of the closer adjacent line segment. The weights for both $ang(L_{n+1})$ and $ang(L_{n-1})$ add up to 1. Therefore, the overall weight sum is 2. The formula for *dir(Car)* is

$$dir(Car) = \frac{1}{2} \cdot \left(ang(L_n) + \frac{d_1}{d_0 + d_1} \cdot ang(L_{n-1}) + \frac{d_0}{d_0 + d_1} \cdot ang(L_{n+1}) \right).$$

3.2.2 Matching: For finding possible locations of a car in another image, we are using the shape-based matching algorithm proposed by (Steger, 2001) and (Ulrich, 2003). The core of this algorithm is visualized in Fig. 9. First, a model image has to be created. This is simply done by cutting out a rectangle of the first image around the car's center. The size of the rectangle is selected in such a way that both car and shadow as well as a part of the surrounding background (usually road) is covered by the area of the rectangle.

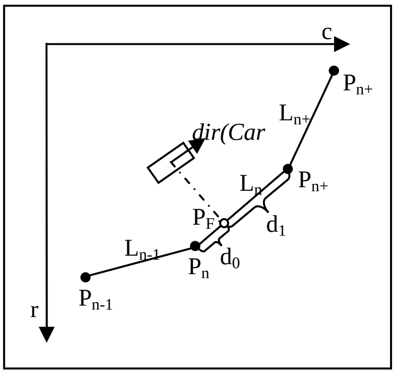

Figure 8. Approximation of the car's motion direction

Still, no other cars or distracting objects such as neighboring meadows should be within the rectangle. The rectangle is oriented in the approximate motion direction that has been calculated before.

A gradient filter is applied to the model image and the gradient directions of each pixel are determined. For run time reasons, only those pixels with salient gradient amplitudes are selected and defined as model edge pixels, in the following also referred as model points. In the RoI of the search image, the gradient filter is also applied. Finally, the model image is matched to the search image by comparing the gradient directions. In particular, a similarity measure is calculated representing the average vector product of the gradient directions of the transformed model and the search image. This similarity measure is invariant against noise and illumination changes but not against rotations and scale. Hence the search must be extended to a predefined range of rotations and scales, which can be easily derived from the motion model and the navigation data. To fulfill real-time requirements also for multiple matches, the whole matching procedure is done using image pyramids. For more details about the shape-based matching algorithm, see (Ulrich, 2003) and (Steger, 2001).

A match is found whenever the similarity measure is above a certain threshold. As a result, we receive the coordinates of the center, the rotation angle, and the similarity measure of the found match. To avoid multiple match responses close to each other, we limited the maximum overlap of two matches to 20%.

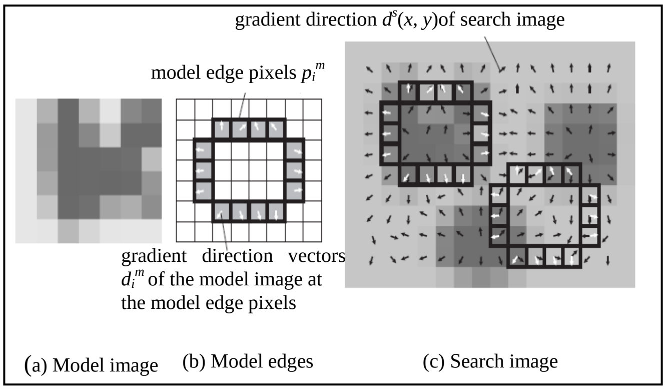

Figure 9. Principle of the shape-based matching method, taken from (Ulrich, 2003), p. 70

3.2.3 Tracking Evaluation

The matching process delivers a number of match positions for M_{12}, M_{23}, and M_{13}. In our tests, we used a maximum number of the 6 best matches for each run. This means that we may receive up to 6 match positions for M_{12} and 36 match positions for M_{23} for each C_1. Also having 6 match positions for M_{13}, we need to evaluate 216 possible tracking combinations for one car. At a first glance, this seems quite cost intensive. Yet, many incorrect matches can be rejected through simple thresholds and consistency criteria so that the computational load can be controlled easily.

Evaluation scheme: As depicted in figure 10, we employ a variety of intermediate weights that are finally aggregated to an overall tracking score. Basically, these weights can be separated into three different categories, each derived from different criteria: *i*) First, a weight for the individual matching runs is calculated (weights w_{12}, w_{23}, and w_{13} in Fig. 10). Here, we consider the single car motion model and the similarity measure as output of the matching algorithm which is also referred to as matching score. *ii*) Based on these weights, a combined weight w_{123} for the combination of the matching runs M_{12} and M_{23} is determined. In this case, the motion consistency is the underlying criterion. *iii*) Finally, weights w_{33} are calculated for the combination of the match positions M_{23} and M_{13}. For a correct match combination, it is essential that the positions of M_{13} and M_{23} are identical within a small tolerance buffer.

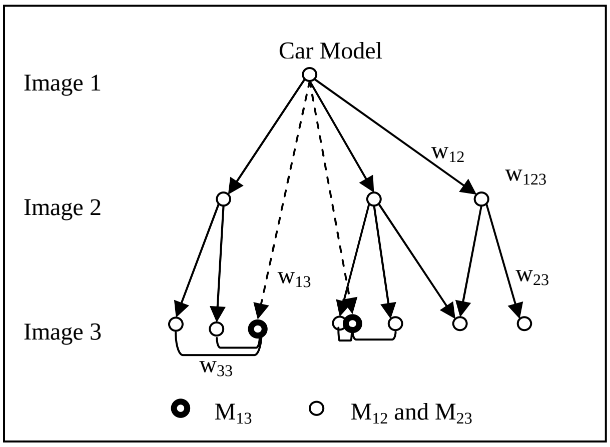

Figure 10. Diagram of the match evaluation process for one car

To avoid crisp thresholds and to allow for the handling of uncertainties, each criterion is mathematically represented as a Gaussian function

$$w(c,\mu,\sigma) = e^{-\frac{(c-\mu)^2}{2\sigma^2}}$$

with parameters mean μ and standard deviation σ evaluating the quality of an observation with respect to the criterion. By this, the weights are also normalized.

In the following, we will outline the calculation and combination of the different weights.

Single Tracking Run: The score w_{match} of the shape-based matching is already normalized (see (Ulrich, 2003) for details). In order to take into account the continuity criterion of a single car's motion, the difference between the motion direction in the first image (say of model C_1) and its conjugate in the next image (say match M_{12}) is considered. In addition, a displacement angle ang_{12} is also included, that essentially reflects the direction difference between the orientation of the trajectory from C_1 to M_{12} and the motion directions in C_1 and M_{12}. From this, the criteria value $Dcross_{12}$ is derived, penalizing across displacements regarding the expected direction $dirS_{12}$. To accommodate the fact that fast cars should move almost straight, $Dcross_{12}$ is multiplied by the distance vel_{12} between M_0 and M_{12}.

$$dirS_{12} = dir(C_1) + \frac{1}{2}\left(dir(C_1) - dir(M_{12})\right)$$

$$Dcross_{12} = \sin\left(dirS_{12} - ang_{12}\right) \cdot vel_{12}$$

The final weight w_{dir} for this criterion is obtained again by measuring its fit with the expected values represented by a Gaussian function. The combined weight w_{12} then calculates to

$$w_{12} = w_{match} \cdot w_{dir} \, .$$

Please note that the formulae above also hold for very slow or even parking cars, since a very small motion distance vel_{12} will scale down $Dcross_{12}$ and thereby allowing for nearly arbitrary direction differences.

Motion consistency: In order to exclude implausible combinations of matches, we examine the consistency of a car's trajectory over image triplets. The first criterion of this category is the change of velocity, i.e. the difference between vel_{12} and vel_{23}.

$$dvel_{13} = \left|vel_{12} - vel_{23}\right| \, .$$

In typical traffic scenarios accelerations of more than $1.5 m/s^2$ rarely happen, while a (nearly) constant speed is common. Again, such values are used to parameterize a Gaussian function resulting in weights w_{vel}.

In order to address the continuity of the trajectory, we carry out the very same calculations as for the single tracking run, now using C_1 and M_{23}, and compare it with the sum of $Dcross_{12}$ and $Dcross_{23}$ of the single displacements. If no difference appears, a car moves totally straight. Deviations from it are again modeled with a Gaussian function, eventually yielding weight w_{dis}. The weights w_{vel} and w_{dis} are combined to w_{123} by multiplication.

$$w_{123} = w_{vel} \cdot w_{dis}$$

Identity of M_{13} and M_{23}: As a last criterion, the identity of Matches M_{13} and M_{23} is checked (see Fig. 10). Weight w_{33} is simply the distance between the match positions of M_{13} and M_{23} put into a Gaussian function.

Final Weight: Assuming that the five individual measurements w_{12}, w_{23}, w_{13}, w_{123}, and w_{33} reflect statistically nearly independent criteria (which, in fact, does not perfectly hold), the final evaluation score W is computed as the product of the five weights:

$$W = w_{12} \cdot w_{23} \cdot w_{13} \cdot w_{33} \cdot w_{123}$$

The correct track is selected as that particular one yielding the best evaluation, however, as long as it passes a lower rejection threshold. Otherwise, it is decided that there is no proper match for a particular car. This may happen when a car is occluded by shadow or another object, but also when it leaves the field-of-view of the images. The latter case can of course be predicted based on a car's previous trajectory. Please note that the track evaluation allows a straightforward extension to more frames or even the tracking of multiple hypotheses if, e.g., the second best track reaches nearly the score of the best track. This option will be included in future work.

4. RESULTS AND DISCUSSION

We tested our algorithm on image triplets. These images have been acquired with a Minolta DiMAGE 7i 5Mpixel camera at a frame rate of 2 Hz. The focal length was approximately 50mm. The approximate flight altitude was between 2000 and 3000 m, therefore we have a ground pixel size of roughly 25-40 cm.

a) b) c)

Figure 11. Results of the tracking in the test image.
a) Detected vehicles in the first image; b) associated cars in the second image; c) final track positions in the third image; see text for explanation of the color coding.

Figure 11 shows the tracking results for one cut-out of an image triplet. It depicts a quite busy highway with cars traveling with different velocities. What makes it also challenging is the presence of the severe shadows on the left carriage way of the highway.

Correctly tracked cars are marked green while incorrect track results are marked red. Black rectangles mark cars which were correctly matched in the second image but moved out of the field-of-view of the third image. Blue marked vehicles are correctly matched in the second image but could not be tracked in the third image even though they were present. In this triplet, 16 out of 20 cars could be correctly tracked. One car moved out of sight in the third image, therefore the comparison with the third image failed. One car was incorrectly tracked. Two cars couldn't be found in the third image although they were present, one of those was at least correctly found in the second image. Note that it is possible that correct and incorrect tracks overlap in the third image. This is the case for the car in front of the yellow bus in Fig. 11. The car itself was tracked correctly, but was also falsely assigned to another car.

In other image triplets with less dominant shadows, correct tracks were found for roughly 90% of the vehicles. However, more testing especially with larger and more variable scenes is still essential. The results reached so far are nonetheless very promising and show the potential of our approach.

The total computation time for all tracks was approximately 5-6 seconds on a 1.8 GHz standard computer. The tracking time for the fourth and following images will further decrease since prior knowledge from the first image triplet can be introduced to better restrain the regions of interest. In addition, we have to mention that the current C++ implementation is by far not yet optimized.

5. FUTURE WORK

As mentioned in Sect. 2 , we want to integrate the tracking approach with an automatic vehicle detection module including lane extraction in the near future. Concerning the tracking, it is planned to apply our approach not only to individual image triplets but – sequentially – also to longer image sequences in order to recover the whole trajectory of each car. Furthermore, when tracking vehicles in longer image sequences, we are planning to extent the motion model by an adaptive component so that besides evaluating the speed and acceleration of a car, the relations to neighboring cars can also be integrated into the evaluation. This would allow a more strict limitation of the search area and deliver a much more precise measure for tracking evaluation. Another area of research would be the detection and integration of context information such as large shadow areas or partial occlusions to be able to also track vehicles that were partially lost during the tracking.

REFERENCES

Dubuisson-Jolly, M.-P., Lakshmanan, S. and Jain, A. (1996): Vehicle Segmentation and Classification Using Deformable Templates. IEEE Trans. on Pattern Analysis and Machine Intelligence 18 (3): 293–308.

Dreschler, L., Nagel, H-H. (1982): Volumetric model and trajectory of a moving car derived from monocular TV frame sequence of a street scene. CGIP, 20: 199-228.

Ernst, I., Hetscher, M., Thiessenhusen, K., Ruhé, M., Börner, A., and Zuev, S. (2005): New approaches for real time traffic data acquisition with airborne systems. Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVI, Part 3/W24, 69-73.

Haag, M. and Nagel, H.-H. (1999): Combination of Edge Element and Optical Flow Estimates for 3D-Model-Based Vehicle Tracking in Traffic Sequences. Int. Journal of Computer Vision 35 (3): 295–319.

Hinz, S. (2005): Fast and Subpixel Precise Blob Detection and Attribution . Proceedings of ICIP 05, Sept. 11-14 2005, Genua.

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

Hinz, S., Baumgartner, A. (2003): Automatic Extraction of Urban Road Nets from Multi-View Aerial Imagery. ISPRS Journal of Photogrammetry and Remote Sensing 58/1-2: 83–98.

Lachaise, M. (2005): Automatic detection of vehicles and velocities in aerial digital image series. Diploma Thesis, Universitee Lyon.

Meffert B, Blaschek R, Knauer U, Reulke R, Schischmanow A, Winkler F (2005): Monitoring traffic by optical sensors. Proc. of Second International Conference on Intelligent Computing and Information Systems (ICICIS 2005): 9-14.

Rajagopalan, A., Burlina, P. and Chellappa, R. (1999): Higher-order statistical learning for vehicle detection in images. Proceedings of the International Conference on Computer Vision, 1999.

Reinartz P., Krauss T., Pötzsch M., Runge H., Zuev S. (2005): Traffic Monitoring with Serial Images from Airborne Cameras. Proc. of Workshop on High-Resolution Earth Imaging for Geospatial Information, Hannover, 2005.

Steger, C. (2001): Similarity measures for occlusion, clutter, and illumination invariant object recognition. In: B. Radig and S. Florczyk (eds.) *Pattern Recognition*, DAGM 2001, LNCS 2191, Springer Verlag, 148–154.

Stilla, U., Michaelsen, E., Soergel, U., Hinz, S., and Ender, J., 2004. Airborne monitoring of vehicle activity in urban areas. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXV, Part B3, 973-979.

Tan, T., Sullivan, G. and Baker, K. (1998): Model-Based Localisation and Recognition of Road Vehicles – International Journal of Computer Vision 27 (1): 5–25.

Toth, C. K., Grejner –Brezinska, D. and Merry, C., 2003. Supporting traffic flow management with high-definition imagery. Proceedings of the Joint ISPRS Workshop on High Resolution Mapping from Space 2003. Hannover, Germany. 6-8 October, (on CDROM).

Ulrich, M., 2003. Hierarchical Real-Time Recognition of Compound Objects in Images. Dissertation, German Geodetic Commission (DGK), Vol. C.