INTEGRATING LOCAL AND GLOBAL FEATURES FOR VEHICLE DETECTION IN HIGH RESOLUTION AERIAL IMAGERY

Stefan Hinz

Chair for Photogrammetry and Remote Sensing, TU München, Arcisstr. 21, 80 333 München – Stefan.Hinz@bv.tum.de

Commission III, WG III/3

KEY WORDS: Vehicle Detection, Vehicle Queues, Urban Areas, Aerial Imagery, Image Analysis

ABSTRACT

This paper introduces a new approach to automatic vehicle detection in monocular high resolution aerial images. The extraction relies upon both local and global features of vehicles and vehicle queues, respectively. To model a vehicle on *local level*, a 3D-wireframe representation is used that describes the prominent geometric and radiometric features of cars including their shadow region. The model is adaptive because, during extraction, the expected saliencies of various edge features are automatically adjusted depending on viewing angle, vehicle color measured from the image, and current illumination direction. The extraction is carried out by matching this model "top-down" to the image and evaluating the support found in the image. On *global level*, the detailed local description is extended by more generic knowledge about vehicles as they are often part of *vehicle queues*. Such groupings of vehicles are modeled by ribbons that exhibit the typical symmetries and spacings of vehicles over a larger distance. Queue extraction includes the computation of directional edge symmetry measures resulting in a symmetry map, in which dominant, smooth curvilinear structures are searched for. By fusing vehicles found using the local and the global model, the overall extraction gets more complete and more correct. In contrast to most of the related work, our approach neither relies on external information like digital maps or site models, nor it is limited to very constrained environments as, e.g., highway scenes. Various examples of complex urban traffic scenes illustrate the applicability of this approach. However, they also show the deficiencies which clearly define the next steps of our future work.

1 INTRODUCTION

This paper deals with automatic detection and counting of cars in high resolution aerial imagery. Research on this topic is motivated from different fields of application: Traffic-related data play an important role in urban and spatial planning, e.g., for road planning and for estimation / simulation of air and noise pollution. In recent years, attempts have been made to derive traffic data also from aerial images, because such images belong to the fundamental data sources in many fields of urban planning. Therefore, an algorithm that automatically detects and counts vehicles in aerial images would effectively support traffic-related analyses in urban planning. Furthermore, because of the growing amount of traffic, research on car detection is also motivated from the strong need to automate the management of traffic flow by intelligent traffic control and traffic guidance systems. Other fields of application are found in the context of military reconnaissance and extraction of geographical data for Geo-Information Systems (GIS), e.g., for site model generation and up-date.

This paper is organized as follows. Section 2 discusses related work on automatic car detection and counting in aerial imagery. The vehicle model underlying our detection algorithm will be developed in Sect. 3. Then, Sect. 4 outlines details of the algorithm and Sect. 5 discusses results achievable with our approach.

2 RELATED WORK

Related work on vehicle detection can be distinguished based on the underlying type of modeling used: Several authors propose the use of an appearance-based, implicit model (Ruskoné et al., 1996, Rajagopalan et al., 1999, Schneiderman and Kanade, 2000, Papageorgiou and Poggio, 2000). The model is created by example images of cars and typically consists of grayvalue or texture features and their statistics assembled in vectors. Detection is then performed by computing the feature vectors from image regions and testing them against the statistics of the model features. The other group of approaches incorporates an explicit model that describes a vehicle in 2D or 3D, e.g., by a filter or wire-frame representation (Burlina et al., 1995, Tan et al., 1998, Haag and Nagel, 1999, Liu et al., 1999, Liu, 2000, Michaelsen and Stilla, 2000, Zhao and Nevatia, 2001, Hinz and Baumgartner, 2001, Moon et al., 2002). In this case, detection relies on either matching the model "top-down" to the image or grouping extracted image features "bottom-up" to construct structures similar to the model. If there is sufficient support of the model in the image, a vehicle is assumed to be detected. Only a few authors model vehicles as part of queues. (Burlina et al., 1997) extract repetitive, regular object configurations based on their spectral signature. In their approach, the search space is limited to roads and parking lots using GIS-information. This seems necessary since the spectrum will be heavily distorted if adjacent objects gain much in influence — even if the spectrum is computed for quite small images patches. In (Ruskoné et al., 1996) and (Michaelsen and Stilla, 2001) vehicle hypotheses extracted by a neural network classifier and a "hot spot detector", respectively, are collinearly grouped into queues while isolated vehicle hypotheses are rejected. Since the queues are not further used to search for missed vehicles, this strategy implies that the vehicle detector delivers an highly oversegmented result, so that grouping is able to separate correct and wrong hypotheses. To the best of our knowledge an approach making use of global and local vehicle features in a *synergetic fashion* for detecting vehicles on downtown streets has not been presented so far.

3 MODEL

3.1 Vehicle Model

Because of the apparent closeness of different objects in urban areas, objects impose strong influence on each other, e.g., trees

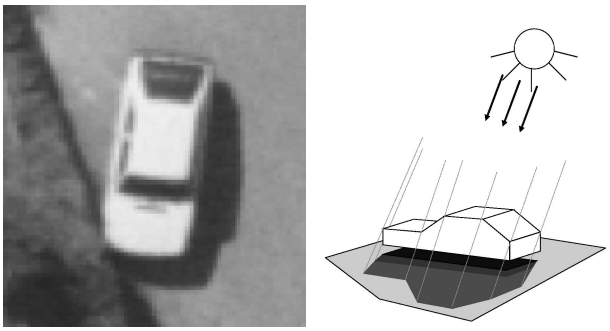


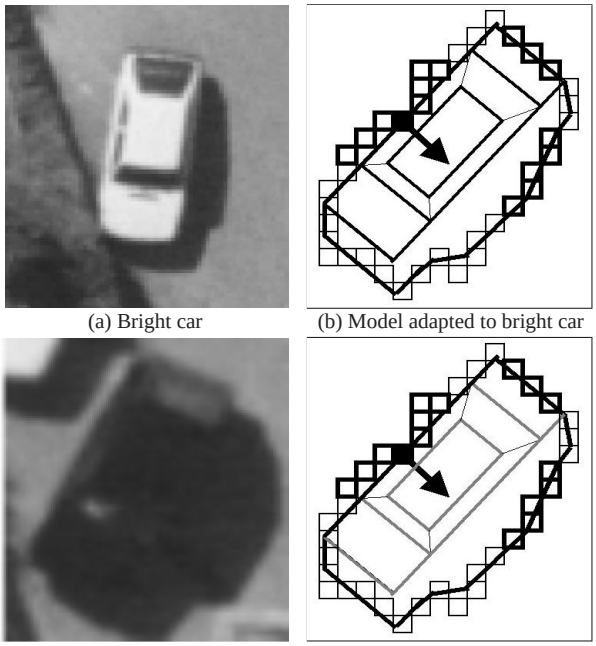
Figure 1: Vehicle model.

may occlude cars partially, buildings cast shadows, materials like glasses or varnish may cause reflections or specularities on cars, etc. Since such influences mostly appear in form of local radiometric disturbances, a model emphasizing a structural description of a vehicle — as an explicit model — seems much more robust than one relying mainly on radiometry as the implicit model does. Another disadvantage of the implicit approach is, that the performance is completely dependent on the training data, while it cannot be assured that the training data capture changes in illumination, viewpoint, and possible influences caused by neighboring objects correctly. In contrast, explicit modeling better allows to focus on the fundamental and robust features of cars and, furthermore, it better allows to employ a hierarchy of levels of detail. However, because of the small size of vehicles, it is clear that a very detailed model is necessary in order to avoid misdetections of objects that are fairly similar to vehicles.

In our approach, we use an explicit model that consists mainly of geometric features and also some radiometric properties. Geometrically, a car is modeled as 3D object by a wire-frame representation. Hence, an accurate computation of the car's shadow projection derived from date, daytime, and image orientation parameters is possible and added to the model. The model further contains substructures like windshield, roof, and hood (see Fig. 1). As radiometric feature, color constancy between hood color and roof color is included. Please note, that color constancy is a relative measure and therefore independent of uniform illumination changes. The only absolute radiometric feature used is the darkness of the shadow region.

The main difference of our vehicle model compared to many other approaches, however, is that the model is adaptive regarding the expected saliency of edge features. Consider, for example, the edge between a car's hood and windshield. In case of a bright car we expect a strong grayvalue edge since a windshield is usually very dark, while in case of a dark car the grayvalue edge may disappear completely. Also the viewing angles relative to the respective vehicle orientation affect the significance of certain edges (see also Fig. 2). To accommodate this, we model the expected saliency of a particular feature depending on vehicle color, vehicle orientation, view point (position in the image), and sun direction. View point and sun direction are derived from the image orientation parameters and vehicle orientation and color are measured from the image.

A disadvantage of the detailed description is, that a large number of models is needed to cover all types of vehicles. To overcome this problem a tree-like model hierarchy may be helpful having a simple 3D-box model at its root from which all models of higher level of detail can be derived subsequently. Such a hierarchy has not been implemented yet.



(a) Bright car (b) Model adapted to bright car

(c) Dark car (d) Model adapted to dark car

Figure 2: Color-adaptive model (aligned to gradient direction): Bold model edges = high expected saliency, gray = low, thin = no.

3.2 Vehicle Queue Model

Due to the high geometric variability of vehicles, it can be hardly assured that the detailed model described above covers all types of vehicles. In some cases, also for a human observer, local features are insufficient to identify a vehicle without doubt (see, e.g. Fig. 2 c). Only provided the contextual information that such a vehicle stands on a road or is part of a queue makes it clearly distinguishable from similar structures. For these reasons our queue model incorporates more generic and more global knowledge. Constraints of the detailed local model are relaxed and, in compensation for this, the global consistency of features is emphasized. More specifically, typical local geometric and radiometric symmetries of vehicles are exploited and, in combination with rough dimensions and spacings of vehicles, they are constrained to form an elongated structure of sufficient length and smoothness (see Fig. 3). In summary following features are used:

- Geometric and radiometric symmetry across queue direction.
- Short, orthogonally intersecting structures characterizing the typical "ladder-like" shape of a vehicle queue.
- Approximately constant width.
- Sufficient length.

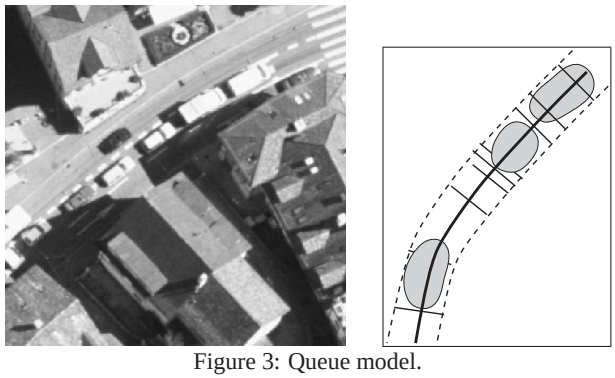
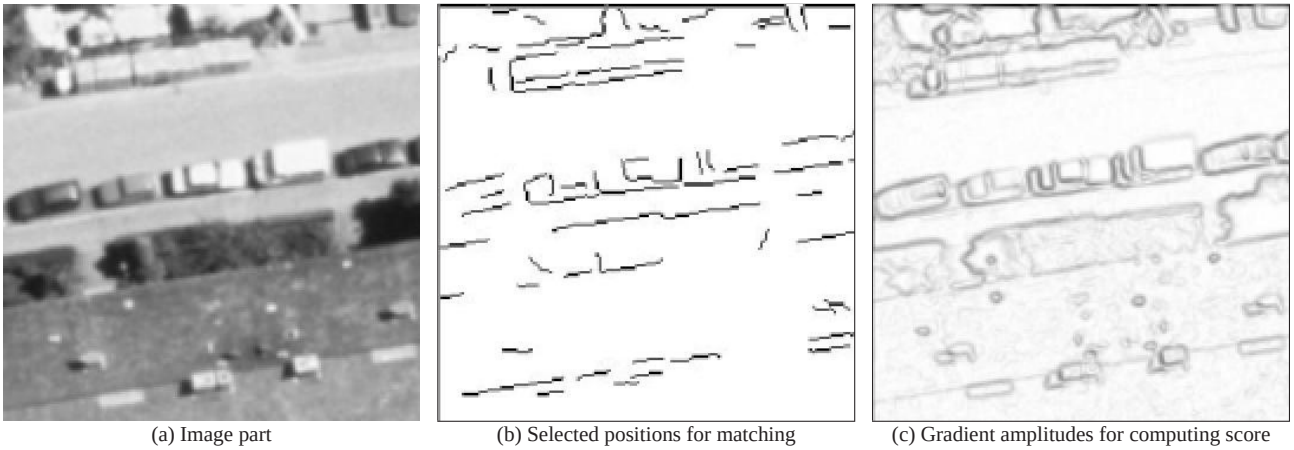


Figure 3: Queue model.



(a) Image part (b) Selected positions for matching (c) Gradient amplitudes for computing score

Figure 4: Intermediate steps during matching

4 DETECTION

To make use of the supplementary properties of the local and global model, following scheme has been implemented: First, the algorithms for vehicle detection (Sect. 4.1) and vehicle queue detection (Sect. 4.2) are run independently. Then, the results of both are fused and queues with enough support from the detailed vehicle detection are selected and further analyzed to recover vehicles missed during vehicle detection (Sect. 4.3). Other detected vehicles, yet not being part of a queue, are kept without deeper analysis.

4.1 Vehicle Detection

Vehicle detection is carried out by a top-down matching algorithm. A comparison with a grouping scheme that groups image features such as edges and homogeneous regions into car-like structures has shown that matching the complete geometric model top-down to the image is more robust. A reason for this is that, in general, bottom-up grouping needs reliable features as seed hypotheses which are hardly given in the case of such small objects like cars (cf. (Suetens et al., 1992)). Another disadvantage of grouping refers to the fact that we must constrain our detection algorithm to monocular images, since vehicles may move within the time of two exposures. Reconstructing a 3D-object from monocular images by grouping involves much more ambiguities than matching a model of the object to the image. The steps of detection can be summarized as follows:

- Extract edge pixels and compute gradient direction.
- Project the geometric model including shadow region to edge pixel and align the model's reference point and direction with the gradient direction, see Fig. 2 b) and d) for illustration.
- Measure reference color / intensity at roof region.
- Adapt the expected saliency of the edge features depending on position, orientation, color, and sun direction.
- Measure features from the image: edge amplitude support of each model edge, edge direction support of each model edge, color constancy, darkness of shadow.
- Compute a matching score (a likelihood) by comparing measured values with expected values.
- Based on the likelihood, decide whether the car hypothesis is accepted or not.

In the following, the evaluation measures involved are explained more in detail. Figures 4 and 5 illustrate the individual steps of matching.

The match of an edge of the wire-frame model with the underlying image is calculated by comparing directional and positional features. Let $\Delta\alpha_i$ be the orientation difference between the gradient ∇I_i at a certain pixel i and the normal vector of the model edge and, furthermore, let d_i be the distance between this pixel and the model edge, then the score S_e [0 ; 1] for the match of a model edge e with n pixels involved is computed by

$$S_e = 1 - \frac{1}{n} \sum_{i=1}^{n} \mid E_e - \frac{1}{2}(A_i + D_i) \mid$$

with

$$A_i = \frac{\pi - \alpha_i}{\pi} \cdot \frac{\| \nabla I_i \|}{c_1} \quad , \qquad D_i = (1 - \frac{d_i}{r}) \cdot \frac{\| \nabla I_i \|}{c_1} \quad ,$$

and with E_e [0 ; 1] being the expected saliency of the model edge, r being the maximum buffer radius around the model edge, and c_1 being a constant to normalize the gradient magnitude $\| \nabla I_i \|$ into the range [0 ; 1]. Finally, the quality of the geometric match of the complete model is calculated as the length-weighted mean of all matching scores S_e. Furthermore, darkness and homogeneity M_s of a shadow region s are evaluated by

$$M_s = \sqrt{\left(1 - \frac{\mu_s}{c_2}\right) \cdot \left(1 - \frac{\sigma_s}{c_3}\right)} \quad ,$$

with μ_s and σ_s being mean and standard deviation of an image region and c_2, c_3 being normalization constants.

To speed up runtime of matching, a number of enhancements and pruning steps have been employed. The most important ones are:

- To avoid redundant computations for projecting models into image space, a database containing all possible (projected) 2D models is created beforehand which is accessed via indices during detection. Since image scale and sun direction are approximately constant for a given scene, the only free parameters are model orientation and x, y position in the image. A reasonable discretization for these variables is derived automatically from image scale and average vehicle size.
- The model is projected only to those positions where edge amplitude has passed a local non-maximum and noise suppression. Though, for calculating the matching score, all pixels are taken into account (see Fig. 4).
- The calculation of features is ordered in such a way, that implausible hypotheses appear yet after a few computations, thus allowing to abort matching immediately.

Figure 5 shows the final result of vehicle detection using the detailed local model.

Figure 5: Detected vehicles: White wire-frame = Vehicles declared as dark during matching and vice-versa.

4.2 Vehicle Queue Detection

Vehicle queue detection is based on searching for one-vehicle-wide ribbons that are characterized by:

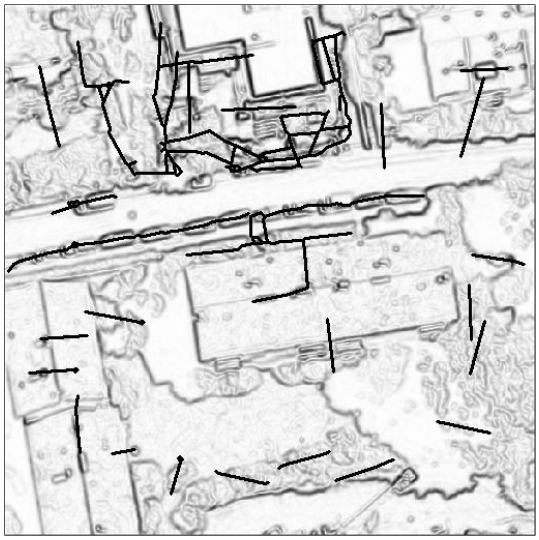
- Significant directional symmetries of grayvalue edges with symmetry maxima defining the queue's center line.
- Frequent intersections of short and perpendicularly oriented edges with homogeneous distribution along the center line.
- High parallel edge support at both sides of the center line.
- Sufficient length.

At first, a "directional symmetry map" is created. The directional symmetry $S_d(i,j)$ of a pixel at position i,j is calculated using a rotating window with a local co-ordinate system u_ϕ, v_ϕ of dimensions $2m+1$ (length) and $2n+1$ (width). For each orientation ϕ of the window, the average symmetry of $2m+1$ cross sections of the gradient amplitude image is computed and, thereafter, the orientation d yielding maximum symmetry is selected, i.e:

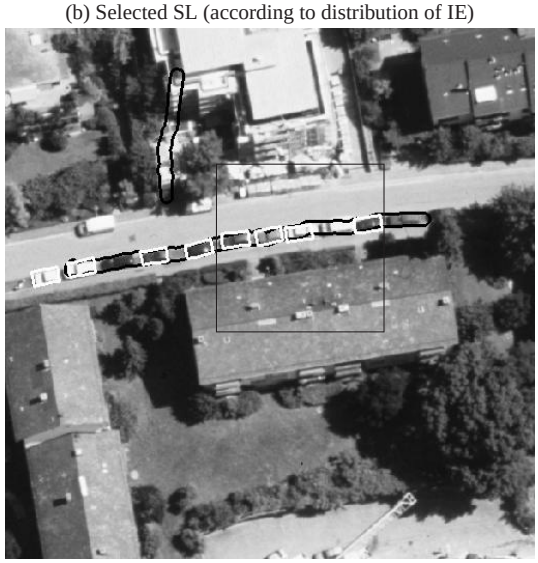
$$S_d = \max_{\substack{\phi= \\ 0\ldots\pi}} \left\{ 1 - \frac{\frac{1}{2n+1}}{(2m+1)c_4} \sum_{\substack{u_\phi= \\ i-m}}^{i+m} \sum_{\substack{v_\phi= \\ j+1}}^{j+n} \left(\|\nabla I_{u_\phi,v_\phi}\| - \|\nabla I_{u_\phi,-v_\phi}\| \right)^2 \right\}$$

with c_4 being a constant to normalize S_d into the range [0 ; 1]. Furthermore, n can be derived from the approximate vehicle width and m is related to the expected minimum length of a straight vehicle queue. Linking adjacent pixels of high symmetry and similar direction into contours yields candidates for queue center lines. These candidates are further evaluated and selected by checking their length and straightness as well as the frequency and distribution of short and orthogonally intersecting edges, i.e., an arbitrary one-vehicle large section of the queue center line must contain at least two intersection points with these edges. The final criterion for selection refers to the edge support found in the gradient amplitude image on each side of the center line at a distance of roughly $\pm n$. Figure 6 illustrates the individual steps of queue extraction.

(a) Symmetry lines (*SL*, black) and intersecting edges (*IE*, white)



(b) Selected SL (according to distribution of IE)



(c) Final queues selected based on parallel edge support (see b); detection using local model is overlayed (white, cf. Sect. 4.1); rectangular box indicates example shown in Fig. 5

Figure 6: Intermediate steps of queue detection.

4.3 Fusion

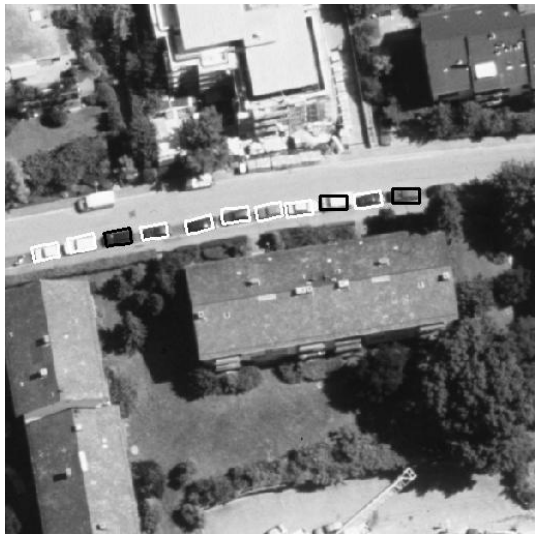
The results of the two independently run algorithms are now fused to make full use of the supplementary properties of the local and global vehicle model. To this end, the results of vehicle detection and queue detection are checked for mutual overlap and parallelism (see Fig. 6 c). A queue is declared as verified if at least one part of it is covered by vehicles found during vehicle detection,

cf. Sect. 4.1. Unverified queues are eliminated from the result. Then, those portions of verified queues that are large enough to enclose a vehicle are analyzed for missing extractions. Since, in many cases, such failures appear through vehicles with weak contrast, an attempt is made to recover these vehicles by extracting homogenous blobs using a regiongrowing algorithm. To get accepted as vehicle, such a blob must almost completely fall into the boundaries of the vehicle queue and the shape parameters of its bounding rectangle must roughly correspond to vehicle dimensions. In case of adjacent blobs, which would cause mutually overlapping vehicles, only the larger one is taken into account (see Fig. 7). Finally, other detected vehicles not being part of a queue are added to the result without further verification. This seems justified since — as a consequence of the stringent local vehicle model employed — the false alarm rate of these results is usually very low.

5 RESULTS AND DISCUSSION

We tested our detection algorithm on a series of high resolution aerial images (ca. $15cm$ ground resolution) of complex downtown areas. No pre-classification of regions of interest has been carried out. The results shown in Figs. 7, 8 and 9 show that nearly all passenger cars have been detected and that the false alarm rate is acceptably low. Also some larger vehicles like vans or small trucks whose geometry deviates from the local model too much have been recovered thanks to the integration of the global queue model. However, such vehicles have been missed throughout all examples whenever they are not part of a queue. This kind of problem could for instance be solved when additional contextual knowledge about roads is available a priori or simultaneously extracted from the image. Failures occur also in regions where the complete road is darkened by building shadows. Similar to the previous case, this could be overcome by pre-classifying shadow regions, so that the vehicle model can be adapted accordingly. Further improvements, mainly regarding the vehicle detection scheme, include the optional incorporation of true-color features and the use of a model hierarchy and/or geometrically flexible models similar to (Olson et al., 1996, Dubuisson-Jolly et al., 1996). The use of multi-view imagery to separate moving from parking vehicles and to estimate vehicle velocity would be another avenue of research.

(a) Queue sections possibly containing cars (black); blobs detected within these sections (white)



(b) Final result: vehicles detected using local model (white) and vehicles recovered through fusion with global model (black)

Figure 7: Fusion of vehicle detection and queue detection.

REFERENCES

Burlina, P., Chellappa, R. and Lin, C., 1997. A Spectral Attentional Mechanism Tuned to Object Configurations. IEEE Transactions on Image Processing 6, pp. 1117–1128.

Burlina, P., Chellappa, R., Lin, C. and Zhang, X., 1995. Context-Based Exploitation of Aerial Imagery. In: IEEE Workshop on Context-based Vision, pp. 38–49.

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

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

Hinz, S. and Baumgartner, A., 2001. Vehicle Detection in Aerial Images Using Generic Features, Grouping, and Context. In: Pattern Recognition (DAGM 2001), Lecture Notes on Computer Science 2191, Springer-Verlag, pp. 45–52.

Liu, G., 2000. Automatic Target Recognition Using Location Uncertainty. PhD thesis, University of Washington, Seattle, WA.

Liu, G., Gong, L. and Haralick, R., 1999. Vehicle Detection in Aerial Imagery and Performance Evaluation. ISL–Technical Report, Intelligent Systems Laboratory, University of Washington, Seattle, WA.

Michaelsen, E. and Stilla, U., 2000. Ansichtenbasierte Erkennung von Fahrzeugen. In: G. Sommer, N. Krüger and C. Perwass (eds), Mustererkennung, Informatik aktuell, Springer-Verlag, Berlin, pp. 245–252.

Michaelsen, E. and Stilla, U., 2001. Estimating Urban Activity on High-Resolution Thermal Image Sequences Aided by Large Scale Vector Maps. In: IEEE/ISPRS joint Workshop on Remote Sensing and Data Fusion over Urban Areas.

Moon, H., Chellappa, R. and Rosenfeld, A., 2002. Performance Analysis of a Simple Vehicle Detection Algorithm. Image and Vision Computing 20(1), pp. 1–13.

Olson, C., Huttenlocher, D. and Doria, D., 1996. Recognition by Matching With Edge Location and Orientation. In: Image Understanding Workshop '96, Morgan Kaufmann Publishers, San Francisco, CA.

Papageorgiou, C. and Poggio, T., 2000. A trainable system for object detection. International Journal of Computer Vision 38(1), pp. 15–33.

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

Ruskoné, R., Guiges, L., Airault, S. and Jamet, O., 1996. Vehicle Detection on Aerial Images: A Structural Approach. In: 13th International Conference on Pattern Recognition, Vol. 3, pp. 900–903.

Schneiderman, H. and Kanade, T., 2000. A Statistical Method for 3D Object Detection Applied to Faces and Cars. In: Computer Vision and Pattern Recognition.

Suetens, P., Fua, P. and Hanson, A., 1992. Computational strategies for object recognition. ACM Computing Surveys 24(1), pp. 5–60.

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

Zhao, T. and Nevatia, R., 2001. Car detection in low resolution aerial image. In: International Conference on Computer Vision.

(a) Vehicle detection (white), queue, and blob detection (black) (b) Vehicles derived from blobs (black)

Figure 8: Example of fusion (white and black boxes in b): Note the successfully recovered vehicles in b) but also the missed vehicles in the central part of the image (due to specularities on cars) and the incorrect extraction in the upper part (due to blob on tree, see a).

(a) (b)

(c) (d)

Figure 9: More results of vehicle detection (queue detection had no influence here): White wire-frame = Vehicles declared as dark during matching and vice-versa. Note the high correctness except for the difficult area shown in d). Reasons for missing extractions — mostly isolated vehicles — are weak contrast, specularities, occlusions, and unmodeled vehicle geometry.