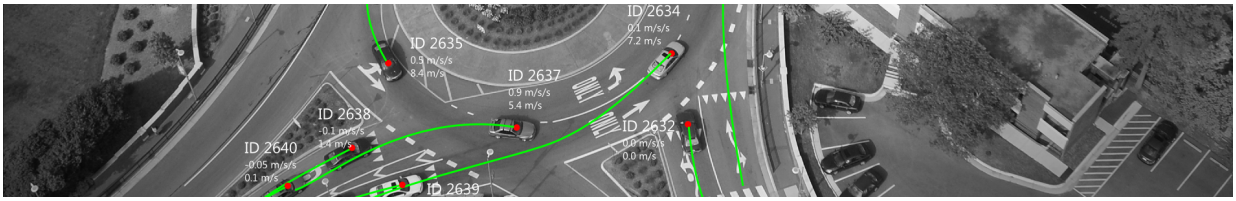
Excel@FIT 2015
http://excel.fit.vutbr.cz

Automatic Vehicle Trajectory Extraction from Aerial Video Data

Adam Babinec*



Abstract
In this paper we present a complex solution to automatic vehicle trajectory extraction from aerial video data, providing a basis for a cost-effective and flexible way to gather detailed vehicle trajectory data in traffic scenes. The video sequences are captured using an action camera mounted on a UAV flying above the traffic scene and processed off-line. The system utilizes video stabilisation algorithm and geo-registration based on RANSAC guided transformation estimation of ORB image feature sets. Vehicles are detected in scene using AdaBoost classifier constructed of Multi-Scale Block Local Binary Patterns features. The vehicle tracking is carried out by multi-target tracker based upon set of intra-independent Bayesian bootstrap particle filters specialized to deal with environmental occlusion, multi-target overlap, low resolution and feature salinity of targets and their appearance changes. The performance of the presented system was evaluated against hand-annotated video sequences captured in distinct traffic scenes. The analysis show promising results with average target miss ratio of 22.5% while keeping incorrect tracking ratio down to 20.4%.

Keywords: Vehicle Detection — Vehicle Tracking — UAV — Aerial Imagery — Particle Filter

Supplementary Material: Demonstration Video

*xbabin02@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Traffic congestion data and information about traffic participants, their trajectory and dynamics during their passage through a traffic scene render to be crucial in numerous applications, from transport infrastructure design analysis and improvement, through analysis of driver's behaviour in various situations (such as unusual intersections, changes in road signs, or weather/lighting conditions) to traffic management (dynamic traffic flow redirection, collision detection, navigation assistance) [1].

The current primary sources of traffic statistics are measurement stations based on induction loops and ultrasonic sensors, which count vehicles that pass a given point on the road. These conventional solutions typically provide data only in the form of basic frequency statistics. Further solution is to employ fixed cameras mounted on ground which can provide additional data, such as vehicle identification and speed estimation. Such solutions however tent to require massive investments and so alternatives are needed. Aerial video surveillance using a wide field-of-view camera sensors offer new opportunities in traffic monitoring, especially in latest years thank to the boom of multicopter UAVs and action cameras.

The utilisation of UAVs operating in low-altitude for traffic inspection has been a major research interest in the past decade; an introduction to the current trends

can be found in this brief survey [2]. Generally, the task of vehicle trajectory extraction can be divided into two essential parts: vehicle detection and vehicle tracking.

Vehicle detection methods can be classified into two categories depending on whether an explicit or implicit model is utilized [3]. Explicit model approach requires a user provided description of detected object – a generic 2D or 3D model of vehicle, which relies on geometric features, such as edges and surfaces of vehicle body or cast shadows, as seen in works of Moon et al. [4] (detection of rectangular body boundaries and windscreens in response of Canny Edge detector), Zhao and Nevatia [5] (adding possibility of shadow presence and incorporating Bayessian network in decision making step) and ZuWhan and Malik [6] (three-dimensional line features based upon detailed model of possible vehicle shapes and probablistic distributions of its dimensions).

Implicit models are derived through collecting statistics over the extracted features, such as Histogram of Oriented Gradients, Local Binary Patterns, specialised pixel-wise features and so forth. The detection for candidate image regions is performed by computing the feature vectors and classifying them against the internal representation usually built up by a cascade classifier training algorithm, such as AdaBoost [7] or more complex dynamic Bayesian networks [8]. The main idea when using implicit models is to use two-stage detection, when in the first stage, the detection candidates are filtered according to various clues such as colour-spatial profiles, density of Harris corners [9], SIFT feature salinity [10] or street extraction techniques [11, 12]. Additionally, temporal information from motion analysis and background subtraction can be incorporated, as seen in [13, 14].

Object tracking algorithms employed in traffic analysis may be divided in two groups: algorithms using Bayesian filters and off-line data association algorithms. One of the most prominent Bayesian filtering algorithm – Kalman filter has been proposed for object tracking as early, as in 1970 [15], and its variations has been used in vehicle tracking in aerial data already in 1993 [16]. To deal with non-linearity of target behaviour model, the extended Kalman filter is suggested by Obolensky in [17]. However, Kalman filter is suited only for tasks modelled by single-modal Gaussian probability density, which is limiting factor in tasks of vehicle tracking in aerial imagery. Therefore, in latest years, particle filter algorithms are being employed, as seen in works of Karlsson and Gustafsson [18] (with its modification using Bayessian Bootstrap algorithm), Samuelsson [19] and Hess et al [20] (pseudo-independent particle filters parametrized by log-linear models with error-driven discriminative filter training). Offline data association tracking algorithms may be based upon graph matching techniques with edges weighted according to spatial proximity and velocity orientation components [21] or kinematic measures, shape and appearance matching [14]. Alternative approaches can be based upon hierarchical connecting of shorter estimations of parts of trajectories – tracklets, into longer ones, eventually forming whole trajectories [22] or maintaining multiple possible candidate tracks per object using a context-aware association (vehicle leading model, avoidance of track intersection) [23].

In this paper we would like to present a system for automatic trajectory extraction from aerial video data. The system utilizes video stabilisation algorithm and geo-registration based on RANSAC [24] guided transformation estimation of ORB image features [25] extracted from annotated reference image and video sequence frames. To produce vehicle detection candidates, we apply background modelling algorithm based on Gaussian Mixture Model [26] which output (foreground mask) is fused together with road reference mask and map of currently tracked vehicles. To keep the system easily modifiable for different target types (pedestrians, animals,...), for vehicle detection, we have utilized implicit model description based on Multi-Scale Block Local Binary Pattern features [27] trained by Viola and Jones's AdaBoost algorithm. The system incorporates two classifiers — *strong classifier*, which detections are considered as significant indication of vehicle presence, and *weak classifier* with higher false alarm rate and very low target miss rate, which output is used to aid vehicle tracking stage. For tracking algorithm we have employed set of fully independent Bayesian bootstrap particle filters [18], one per each target. The algorithm was modified to cope with nature of aerial video data — environmental occlusion, multi-target overlap, low resolution and feature salinity of targets and their appearance changes.

2. Vehicle Detection

The purpose of vehicle detection stage is to provide new targets for tracking stage and aid tracking stage by giving clues about the positions of already tracked vehicles. The preceding step of vehicle detection is transformation of input image into the real world coordinate system, removing perspective effect. This transformation is derived from known transformation of reference image into real world coordinate system

and estimated transformation between reference image and current image. This step leads to orthographic representation of the scene, reducing the range of possible candidate sizes. Candidate generation is limited to road surface area retrieved from annotated reference image and the candidate must fulfil at least one of the following conditions:

- Center of candidate area exhibits the signs of motion. To detect the motion, we employed background cubtration method based on Gaussian Mixture Model as presented in [26].
- Candidate area is overlapped significantly by currently tracked vehicle. To test this condition, the position of the tracked vehicle has to be predicted – using the motion model of vehicle as described in section 3.2.

For the selection of appropriate features and construction of a robust detector, we have utilized Viola and Jones's Adaboost algorithm [7], but instead of HAAR features, we have employed Multi-Scale Block Local Binary Patterns (MB-LBP) features, which calculation is computationally less expensive, and due to the ability to encode both microstructures and macrostructures of the image area, they are shown to be more robust to illumination changes [28] and have significantly smaller false alarm rate than HAAR features, while keeping comparable hit rate [28].

The classifier was trained on hand annotated training dataset with 20000 positive and 20000 negative samples taken from aerial videos. The size of every sample is 32×32 pixels. Positive samples contain motor vehicles of different types, colours and orientations. The negative samples were created from surroundings of the intersections, as well as from the road surface with the emphasis on horizontal traffic signs. The resulting trained classifier consists of 18 stages of boosting cascade of simple MB-LBP classifiers. For the further improvement of detection assistance for tracking algorithm, the trained detector was split to two classifiers:

- *strong vehicle classifier* – consisting of all 18 stages of trained classifier, having small false alarm rate. The detections signalled by this classifier, further referenced as *strong detections*, are considered as very significant indication of vehicle presence in detection candidate area, and are treated as such further in tracking stage of the algorithm, especially as basis for new tracking targets.
- *weak vehicle classifier* – consisting of first 11 stages of trained classifier, having higher false alarm rate. This classifier is more benevolent than *strong vehicle classifier* accepting much more detection candidates and its output is used to aid vehicle tracking, acting as tracking attractors. These detections, further referenced as *weak detections*, form basis for *heat function*.

2.1 Heat Function

To speed up the operations with *weak detections*, we proposed following function which for given point in two-dimensional real-world coordinate system returns the significance of this point according to the set of *weak detections* :

$$h(\mathbf{x}) = 1 + \sum_{d \in \mathscr{D}_{weak}} f(\mathbf{x}, \mathbf{x}_d, \sigma_d) \quad , \tag{1}$$

where $\mathscr{D}_{weak}$ is a set of current *weak detections*, $\mathbf{x}_d$ is a position of the detection d in real-world coordinate system, σ_d is size of detection d in real-world coordinate system, $f(\mathbf{x}, \mu, \sigma)$ represents the value of a multivariate normal distribution $\mathscr{N}(\mu, \Sigma)$ expressed at point $\mathbf{x}$. Matrix Σ is constructed as follows:

$$\Sigma = \begin{bmatrix} (0.16\sigma)^2 & 0 \\ 0 & (0.16\sigma)^2 \end{bmatrix} \quad . \tag{2}$$

3. Vehicle Tracking

Tracking of vehicle targets in the scene is carried out using the detections generated by both strong and weak vehicle classifiers and data extracted from geo-registered input frame. It is divided into three steps: Detection-Track Association, Tracks Update and Tracking Termination, which all will be described in following subsections.

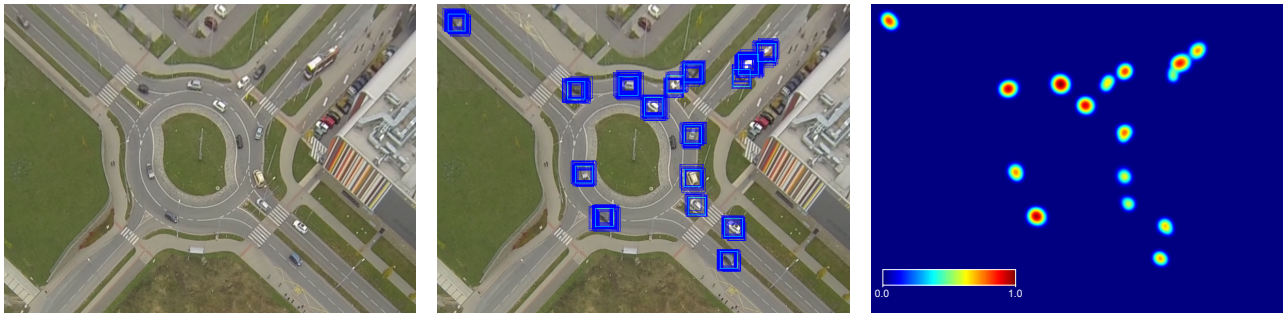
3.1 Detection-Track Association

For each detection from both sets of weak and strong detections, the best fitting tracked object is found and vice versa, while strong detections being favoured. Formally, let $\mathscr{D}_{weak}$ be a set of weak detections, $\mathscr{D}_{strong}$ be a set of strong detections, $\mathscr{D} = \mathscr{D}_{weak} \cup \mathscr{D}_{strong}$ be a set of all detections, $\mathscr{T}$ be a set of all currently tracked objects and $\mathscr{A}$ is set of constructed associations. Then for every constructed association $(d,t) \in \mathscr{A}$, where $d \in \mathscr{D}$ and $t \in \mathscr{T}$, it must be that:

$$\begin{aligned} \forall d' \in \mathscr{D}, d' \neq d : \operatorname{diff}(d',t) \geq \operatorname{diff}(d,t) \vee \\ d \in \mathscr{D}_{strong} \wedge d' \in \mathscr{D}_{weak} \end{aligned} \tag{3}$$

and

$$\begin{aligned} \forall t' \in \mathscr{T}, t' \neq t : \operatorname{diff}(d,t') \geq \operatorname{diff}(d,t) \wedge \\ \operatorname{diff}(d,t) \leq diff_{max} \quad , \end{aligned} \tag{4}$$



(a) Geo-registered input image

(b) Overlaid detections: dark blue rectangles represent weak detections, light blue rectangles represent strong detections.

(c) Colourised representation of heat function values normalised into interval $< 0, 1 >$.

Figure 1. Example of *heat function* defined in equation (1) constructed from a set of weak detections.

where $diff_{max}$ is arbitrary constant and $\mathrm{diff}(d,t)$ is function, which returns the difference of rectangles representing detection d and tracked object t. It is defined as follows:

$$\mathrm{diff}(d,t) = \frac{|r_d| + |r_t| - 2|r_d \,\&\, r_d|}{|r_d| + |r_t|} \quad , \tag{5}$$

where r_d is rectangle representing detection d, r_t is rectangle representing tracked object t, function $|r|$ returns area of rectangle r and binary operator & represents intersection of given operands.

Every strong detection that has not been associated to any of currently tracked objects forms basis for new tracking target – its initial size, position and target model is derived from detection, while velocity is set to zero.

3.2 Tracks Update

Tracks update step engages tracking algorithm itself to estimate the position of tracked targets in the current image, according to sequence of all video frames up to this time moment. For this purpose, a set of intra-independent Bayesian bootstrap particle filters has been employed – one per each target, similarly as in [29, 30]. Bootstrap filter uses transition density as a proposal density and performs resampling step in each iteration [18].

For the algorithm to be able to track its target, the description of tracked object is necessary – target model. Our system uses a rectangular descriptor template derived from detection area of the target, consisting of 3 colour channels (RGB) and edge map channel – sum of absolute response of Scharr operator in both x and y directions in the image. This template is extracted from geo-registered image bounded by area defined by initial detection, and it is resized to uniform size 32×32 pixels. To achieve the plasticity of target model, the template $\mathbf{T}_t$ of tracked object t is updated if one of the following events happen:

- There is currently associated strong detection to object t.
- There is currently associated weak detection to object t and the value of *heat* function $h(\mathbf{x})$ as described in equation (1), evaluated at the estimated position of tracked object, is greater than threshold T_{heat}.

In the case of template update, the values of template are altered by weighted average of the former template and new template extracted from currently processed image, where former template has weight of 0.95 and new template has weight of 0.05. Additionally, to prevent undesirable swaps between targets, the template update is disabled if multiple targets overlap.

The particle filter uses particles which states are defined by target position vector $\mathbf{x}$, target velocity vector $\mathbf{v}$ and target size s, all in real-world coordinate system, forming together particle state vector:

$$\mathbf{p} = \begin{bmatrix} \mathbf{x}(0) \\ \mathbf{x}(1) \\ \mathbf{v}(0) \\ \mathbf{v}(1) \\ s \end{bmatrix} \quad . \tag{6}$$

As the transition model, we consider target position is an integration of target velocity, and therefore it is represented by following matrix:

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad , \tag{7}$$

and state transition equation:

$$\mathbf{p}^{(i+1)} = \mathbf{D}\left(\mathbf{p}^{(i)} + \mathbf{n}\right) \quad , \tag{8}$$

where $\mathbf{n}$ is 5D noise vector, which elements are generated randomly by normal distribution $\mathcal{N}\left(\mu, \sigma^2\right)$,

where $\mu = 0$ and value of σ is user defined parameter for each element of noise vector. In case of velocity noise and size noise, the parameter σ is constant during the whole course of tracking, whilst for position noise, the value of parameter σ evolves according to following equation:

$$\sigma_{\mathbf{x}}^{(i)} = \sigma_{\mathbf{x}} \left(1 + f^i m\right) \quad , \tag{9}$$

where, i is sequential order of currently processed image from target's tracking inception, $\sigma_{\mathbf{x}}$ is target value of $\sigma_{\mathbf{x}}^{(i)}$ for $i \to \infty$, f is falloff ratio and m is initial multiplier of covariance. The application of this approach causes the position of particle to be affected more by random noise than by its velocity during the early stage of tracking, and afterwards slowly elevating the effect of velocity. This way, the particle's velocity may slowly adapt while elevating its effect on particle behaviour.

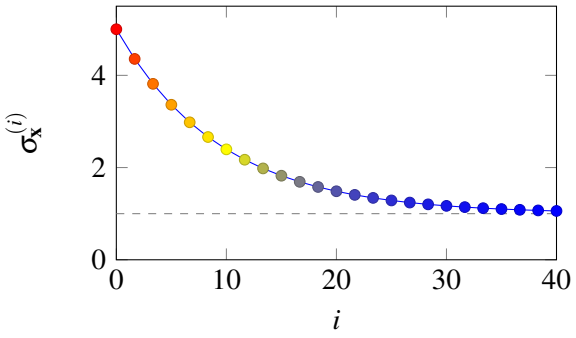


Figure 2. Graph of function $\sigma_{\mathbf{x}}^{(i)}$ defined by equation (9) according to values of i for parameters $\sigma_{\mathbf{x}} = 1$, $f = 0.9$ and $m = 4$.

The evaluation and resampling step of the particle filter is based upon importance weight $\mathscr{W}(p,t)$ of particle p for tracked target t which is defined as:

$$\mathscr{W}(p,t) = e^{App(p,t)^2 \cdot Att(p)} \quad . \tag{10}$$

The appearance similarity function $App(p,t)$ of particle p to target t is evaluated as follows:

$$App(p,t) = \frac{1}{1 + \mathrm{SAD}_C(T_t, T_p)} \quad , \tag{11}$$

where T_t is model template of target object t, T_p is model template of fictitious target object based on particle p and $\mathrm{SAD}_C(T_1, T_2)$ is sum of absolute differences of templates T_1 and T_2 across all their channels spatially weighted by circular mask around the centre of the templates at point $c = (16, 16)\ px$, with radius of $16\,px$. The attraction factor function $Att(p)$ of particle p is defined as:

$$Att(p) = h(\mathbf{x}_p) \quad , \tag{12}$$

where $h(\mathbf{x})$ is *heat function* as defined in equation (1) and $\mathbf{x}_p$ is position of particle p in real world coordinate system. The estimated state $\mathscr{E}(t)$ of the target t is represented as the highest-weighted particle (*maximum a posteriori*), i.e.:

$$\mathscr{E}(t) = \arg\max_{p \in \mathscr{X}_{(t)}} \left(\mathscr{W}(p,t)\right) \quad , \tag{13}$$

where $\mathscr{X}_{(t)}$ is set of all particles of particle filter modelling tracked target t. Resampling step of particle filter for particle set $\mathscr{X}_{(t)}$ is carried out using weight proportionate random selection, also known as roulette wheel principle, according to values of particle's importance weight.

3.3 Tracking Termination

Tracking of target $t^\dagger$ is terminated when one of the following conditions are met:

- Target $t^\dagger$ leaves the area defined by annotated road surface.
- Target model of target $t^\dagger$ has not been updated for certain amount of time steps.
- Target t is overlapping with another target t' for certain amount of time steps and following condition is met:

$$App\left(p^*_{t'}, t'\right) > App\left(p^*_{t^\dagger}, t^\dagger\right) \quad , \tag{14}$$

  where $App(p,t)$ is appearance similarity function as described in equation (11), $p^*_{t'}$ is highest weighted particle of particle set $\mathscr{X}_{(t')}$ tied with target t' and $p^*_{t^\dagger}$ is highest weighted particle of particle set $\mathscr{X}_{(t^\dagger)}$ tied with target $t^\dagger$.

In case of tracking termination of target $t^\dagger$, its generated trajectory is analysed. If it is found that the target $t^\dagger$ has fully entered the analysed area of the scene, passed through it and left it, in that order, it is considered as successful tracking. Otherwise, the tracking is considered as unsuccessful and is rejected.

4. Experiments

The system presented in this paper has been evaluated on two sequences of video data captured by action camera GoPro Hero3 Black Edition mounted on a UAV flown at the height of approx. 100 m above the road surface. The video was captured with the resolution of 1920 px × 980 px at 29 Hz. Due to utilization of ultra-wide angle lens, the diagonal field of view was 139.6°. The spatial resolution of the captured scene was approximately 10.5 cm/px. In the course of data acquisition the UAV was stabilized around a fixed position in the air.

The first sequence was captured near Netroufalky construction site in Bohunice, Brno, Czech Republic. The second sequence was captured at the site of roundabout junction of Hamerska road and Lipenska road near Olomouc, Czech Republic.

(a) Netroufalky

(b) Olomouc

Figure 3. Scenes used for evaluation.

The evaluation was carried out against ground truth annotated by hand consisting of trajectories of all vehicles that both fully entered and exited the crossroad area during the evaluation sequence. As high level evaluation metrics we used relative number of missed targets $\mathbf{NMT_r} = \frac{\mathbf{NMT}}{|L|}$, relative number of false tracks $\mathbf{NFT_r} = \frac{\mathbf{NFT}}{|L|}$, average number of swaps in tracks **ANST** and temporal average of measure of completeness $\mathbf{MOC_a}$ which is defined as follows:

$$\mathbf{MOC_a} = \frac{\sum_{k=0}^{n_{video}} Comp(k)}{n_{video}}. \quad (15)$$

Metrics **NMT** and **NFT** are defined as in [31], averaged by arithmetic mean across the whole evaluation sequence, $|L|$ is a number of ground truth tracks, n_{video} is the number of images in the evaluation sequence, **ANST** and $Comp(k)$ are described in [31] as well.

The spatial precision of the algorithm was evaluated using root mean square error (**RMSE**) of track position averaging over all valid tracks. The estimated track $\mathscr{E}_i$ is considered corresponding to the ground truth l_j at given time moment k if the track $\mathscr{E}_i$ is the nearest track to the truth l_j at the moment k and vice versa, and the distance between them is less than threshold $t_{dist} = 3\,\mathrm{m}$. The estimated track is valid if it corresponds to the same ground truth for the whole period of the vehicle presence in the analysed area.

Table 1. Results of the evaluation.

Sequence	Netroufalky	Olomouc
True Tracks #	165	168
Estimated Tracks #	173	153
Valid Estimated Tracks #	143	115
$\mathbf{NMT_r}$	0.054	0.197
$\mathbf{NFT_r}$	0.052	0.163
ANST	0.049	0.066
$\mathbf{MOC_a}$	0.841	0.667
RMSE $[m]$	1.045	1.102

Incorrect or missed targets in the sequence can be easily noticed and fixed by a human user. To indicate the necessary effort that is needed to fix the invalid tracks the following graph shows the dependency of the number of missed true tracks (true tracks that were not assigned to any valid estimated track) on the rate of adjustments needed to the best partially matching estimated tracks.

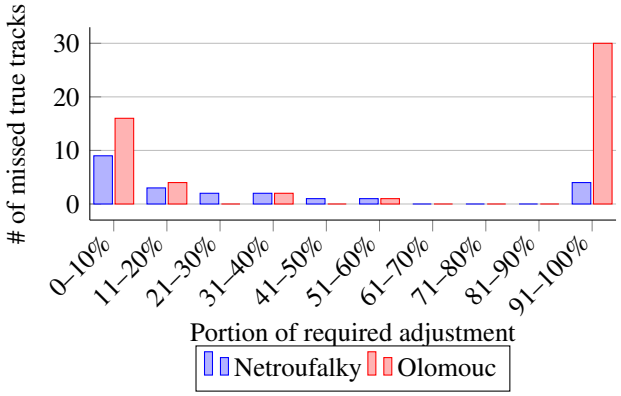


Figure 4. Histogram of missed true tracks, according to required adjustment to best matching fragments.

5. Conclusions

In this paper, we have presented a system for vehicles' trajectories extraction from aerial video data captured by a UAV. The most important part of the system is its tracking algorithm based upon set of intra-independent Bayesian bootstrap particle filters which were modified

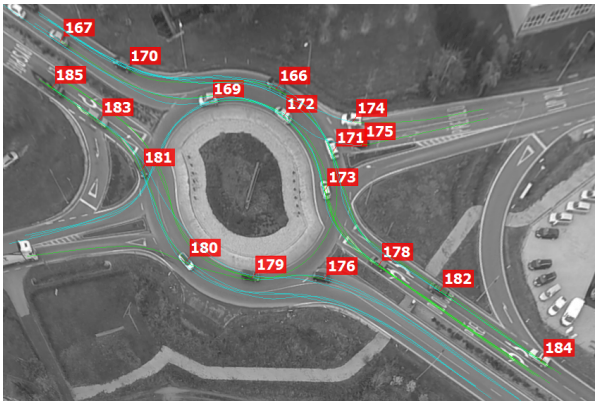


Figure 5. Output example of the presented system.

to deal with special caveats which are derived from the nature of vehicle tracking in aerial video data - huge data amount, low spatial resolution and temporal variance of camera position and tracked objects.

The system's accuracy and performance is at present stage not suitable for wide application, but we believe, that it can be improved further. Both the detection algorithm and tracking algorithm were implemented with plans of further improvements and possible specialisation. The detection cascades can be retrained for different object types and their performance improved by collecting much more training data. Alternatively, we consider change to deep learning methods for target detection, as they render to be superior when huge set of training data is provided [32]. Also, the tracking algorithm can be improved by introducing intra-dependency, driver model, target shape estimation, occlusion ordering and other usual approaches in multi-object tracking. Further discussion, improvements and tests will be part of my master thesis.

The potential of the system is also indicated by the fact, that it is already being used as assisting tool in process of intersection analysis and design at Institute of Road Structures under Faculty of Civil Engineering, Brno University of Technology.

Acknowledgement

This work was done as part of master thesis under supervision of Ing. Jaroslav Rozman, PhD.[1] and assistance of Ing. David Herman[2]. I would also like to thank Ing. Jiří Apeltauer[3] for providing the necessary video sequences for evaluation and development.

[1]Faculty of Information Technology, Brno University of Technology

[2]RCE Systems, s.r.o., Brno, Czech Republic

[3]Faculty of Civil Engineering, Brno University of Technology

References

[1] *Highway Capacity Manual: Practical Applications of Research*. U.S. Dept. of Commerce, Bureau of Public Roads, 2000.

[2] J.-N. Lee and K.-C. Kwak. A trends analysis of image processing in unmanned aerial vehicle. *International Journal of Computer, Information Science and Engineering*, 8(2):2 – 5, 2014.

[3] T.T. Nguyen, Grabner H., Gruber B., and Bischof H. On-line boosting for car detection from aerial images. In *IEEE International Conference on Reasearch, Innovation and Vision for the Future*, pages 87–95, 2007.

[4] H. Moon, R. Chellapa, and A. Rosenfeld. Performance analysis of a simple vehicle detection algorithm. *Image and Vision Computing*, 20:1 – 13, January 2002.

[5] T. Zhao and R. Nevatia. Car detection in low resolution aerial images. In *Eight IEEE International Conference on Computer Vision*, pages 710–717, Vancouver, 2001. IEEE.

[6] K. ZuWhan and J. Malik. Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking. In *Proceedings on Ninth IEEE International Conference on Computer Vision*, pages 524–531, October 2003.

[7] Y. Freund and R.E. Schapire. A short introduction to boosting. In *Journal of Japanese Society for Artificial Intelligence*, pages 771–780, 1999.

[8] K.P. Murphy. *Dynamic bayesian networks: representation, inference and learning*. PhD thesis, University of California, Berkeley, 2002.

[9] J. Gleason, A.V. Nefian, X. Bouyssounousse, T. Fong, and G. Bebis. Vehicle detection from aerial imagery. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2065–2070, May 2011.

[10] T. Moranduzzo and F. Melgani. Automatic car counting method for unmanned aerial vehicle images. *IEEE Transactions on Geoscience and Remote Sensing*, 52(3):1635–1647, March 2014.

[11] G. Pacher, S. Kluckner, and H. Bischof. An improved car detection using street layer extraction. In *Proceedings of the 13th Computer Vision Winter Workshop*, Ljubljana, 2008.

[12] S. Tuermer, J. Leitloff, P. Reinartz, and U. Stilla. Automatic vehicle detection in aerial image sequences of urban areas using 3d hog features.

Photogrammetric Computer Vision and Image Analysis, 28:50–54, 2010.

[13] S. Tuermer, J. Leitloff, P. Reinartz, and U. Stilla. Motion component supported boosted classifier for car detection in aerial imagery. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science*, 2011.

[14] J. Xiao, H. Cheng, H. Sawhney, and F. Han. Vehicle detection and tracking in wide field-of-view aerial video. In *2010 IEEE Conference on Computer Vision and Pattern Recognition*, pages 679–684, June 2010.

[15] R.A. Singer. Estimating optimal tracking filter performance for manned maneuvering targets. *IEEE Transactions on Aerospace and Electronic Systems*, AES-6(4):473–483, July 1970.

[16] D. Koller, J. Weber, and J. Malik. Robust multiple car tracking with occlusion reasoning. Technical Report UCB/CSD-93-780, EECS Department, University of California, Berkeley, Berkeley, November 1993.

[17] N. Obolensky. Kalman filtering methods for moving vehicle tracking. Master's thesis, University of Florida, 2002.

[18] R. Karlsson and F. Gustafsson. Monte carlo data association for multiple target tracking. In *Target Tracking: Algorithms and Applications*, volume 1, pages 13/1–13/5 vol.1, October 2001.

[19] O. Samuelsson. *Video Tracking Algorithm for Unmanned Aerial Vehicle Surveillance*. PhD thesis, KTH Royal Institute of Technology, School of Electrical Engineering, 2012. Master's Degree Project.

[20] R. Hess and A Fern. Discriminatively trained particle filters for complex multi-object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 240–247, June 2009.

[21] Idrees H. Reilly V. and Shah M. Detection and tracking of large number of targets in wide area surveillance. In *11th European Conference on Computer Vision*, volume 6313, pages 186–199, Berlin, 2010. Springer-Verlag Berlin Heidelberg.

[22] J. Prokaj, X. Zhao, and G. Medioni. Tracking many vehicles in wide area aerial surveillance. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 37–43, June 2012.

[23] I. Saleemi and M. Shah. Multiframe many-many point correspondence for vehicle tracking in high density wide area aerial videos. *International Journal of Computer Vision*, 104(2):198–219, 2013.

[24] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communication ACM*, 24(6):381–395, June 1981.

[25] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An Efficient Alternative to SIFT or SURF. In *International Conference on Computer Vision*, Barcelona, 2011. Willow Garage.

[26] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 2, pages 28–31 Vol.2, August 2004.

[27] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S.-Z. Li. Learning multi-scale block local binary patterns for face recognition. In *Advances in Biometrics*, volume 4642 of *Lecture Notes in Computer Science*, pages 828–837. Springer Berlin Heidelberg, 2007.

[28] Y. Ju, H. Zhang, and Y. Xue. Research of Feature Selection and Comparison in AdaBoost based Object Detection System. Number 22, 2013.

[29] R. Danescu, F. Oniga, S. Nedevschi, and M. Meinecke. Tracking multiple objects using particle filters and digital elevation maps. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 88–93, June 2009.

[30] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *IEEE International Conference on Robotics and Automation.*, volume 2, pages 1665–1670 vol.2, 2001.

[31] A. A. Gorji, R. Tharmarasa, and T. Kirubarajan. Performance measures for multiple target tracking problems. In *Proceedings of the 14th International Conference on Information Fusion*, July 2011.

[32] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2155–2162. IEEE, 2014.