Optical Engineering

OpticalEngineering.SPIEDigitalLibrary.org

Moving object detection in top-view aerial videos improved by image stacking

Michael Teutsch
Wolfgang Krüger
Jürgen Beyerer

SPIE.

Michael Teutsch, Wolfgang Krüger, Jürgen Beyerer, "Moving object detection in top-view aerial videos improved by image stacking," *Opt. Eng.* **56**(8), 083102 (2017), doi: 10.1117/1.OE.56.8.083102.

Moving object detection in top-view aerial videos improved by image stacking

Michael Teutsch,[a] Wolfgang Krüger,[a] and Jürgen Beyerer[a,b,*]
[a]Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, Karlsruhe, Germany
[b]Institute for Anthropomatics and Robotics IAR, Karlsruhe Institute of Technology KIT, Karlsruhe, Germany

Abstract. Image stacking is a well-known method that is used to improve the quality of images in video data. A set of consecutive images is aligned by applying image registration and warping. In the resulting image stack, each pixel has redundant information about its intensity value. This redundant information can be used to suppress image noise, resharpen blurry images, or even enhance the spatial image resolution as done in super-resolution. Small moving objects in the videos usually get blurred or distorted by image stacking and thus need to be handled explicitly. We use image stacking in an innovative way: image registration is applied to small moving objects only, and image warping blurs the stationary background that surrounds the moving objects. Our video data are coming from a small fixed-wing unmanned aerial vehicle (UAV) that acquires top-view gray-value images of urban scenes. Moving objects are mainly cars but also other vehicles such as motorcycles. The resulting images, after applying our proposed image stacking approach, are used to improve baseline algorithms for vehicle detection and segmentation. We improve precision and recall by up to 0.011, which corresponds to a reduction of the number of false positive and false negative detections by more than 3 per second. Furthermore, we show how our proposed image stacking approach can be implemented efficiently. [DOI: 10.1117/1.OE.56.8.083102]

Keywords: visual surveillance; unmanned aerial vehicles; video processing; object detection; image stacking.

Paper 170239 received Feb. 20, 2017; accepted for publication Jul. 6, 2017; published online Aug. 16, 2017.

1 Introduction

Aerial videos acquired by airplanes or fixed-wing unmanned aerial vehicles (UAVs) are a good source of data for ground surveillance. Such kind of data can be used for applications, such as automatic traffic monitoring, border surveillance, or protection of critical infrastructures. In this article, we will focus on video data acquired with a frame rate of 25 Hz by a small UAV in top-down view at an altitude of 400 m. The ground coverage is up to $0.5\ \mathrm{km}^2$ with a ground sampling distance (GSD) of about $0.3\ \mathrm{m/pixel}$. Such parameters are usual for so-called full-motion video (FMV) sequences. Additionally, there is one specific property that we can find in top-view videos with small-sized objects on the ground that are observed; there is no significant change in appearance of the objects in the image over time. This is crucial for the successful application of image stacking.

Image stacking is a well-known method that is used to improve the quality of images in video data. A set of consecutive images is aligned by applying image registration and warping.[1] In the resulting image stack, each pixel position contains redundant information about its intensity or color value. This redundant information can be used to perform image fusion to suppress image noise,[2,3] handle motion blur,[4–6] detect independent motion in images using background models,[7,8] or even enhance the spatial image resolution as done in multiframe super-resolution.[9–12] However, the performance of the above-mentioned methods is usually dependent on the assumption that the scene is static and that the viewing angle does not change significantly over time. At the same time, moving objects can get blurred or distorted as the images inside the image stack are registered and aligned with respect to (w.r.t.) the stationary background. Hence, moving objects need to be handled explicitly as this is done for super-resolution.[13,14]

In this article, we apply image stacking in an innovative way: image alignment is not performed w.r.t. the stationary background but to small moving objects instead. In our aerial FMV data, moving objects are mainly cars but also other vehicles on the ground such as motorcycles. Then, we apply a pixelwise temporal filter (e.g., a median filter as done in Ref. 3) to the image stack to remove noise and compression artifacts from the moving objects. More interestingly, we intentionally add motion blur to the background that surrounds the moving objects. This effect is visualized in Fig. 1. We aim at detecting the moving vehicle that is marked with a red cross. As we can see in the zoomed original image (left box inside each image), there are potential distractors, such as parked and overtaking vehicles. By applying our proposed image stacking method (right box inside each image), it takes only 28 images (∼1 s) to blur the distractors effectively. Although camera and objects are moving, the observed vehicle's appearance is nearly constant over time due to the top-view camera angle and due to the large distance between camera and scene. In this way, stationary image content is removed that can modify the observed object's appearance and, thus, can disturb the detection and segmentation process, such as (1) partial occlusions by trees, power supply lines, or buildings, (2) stationary objects close to the observed object, such as parked vehicles or buildings, and (3) street textures, such as cobblestones or road markings. To demonstrate the effectiveness of the proposed approach, two baseline

*Address all correspondence to: Jürgen Beyerer, E-mail: juergen.beyerer@iosb.fraunhofer.de

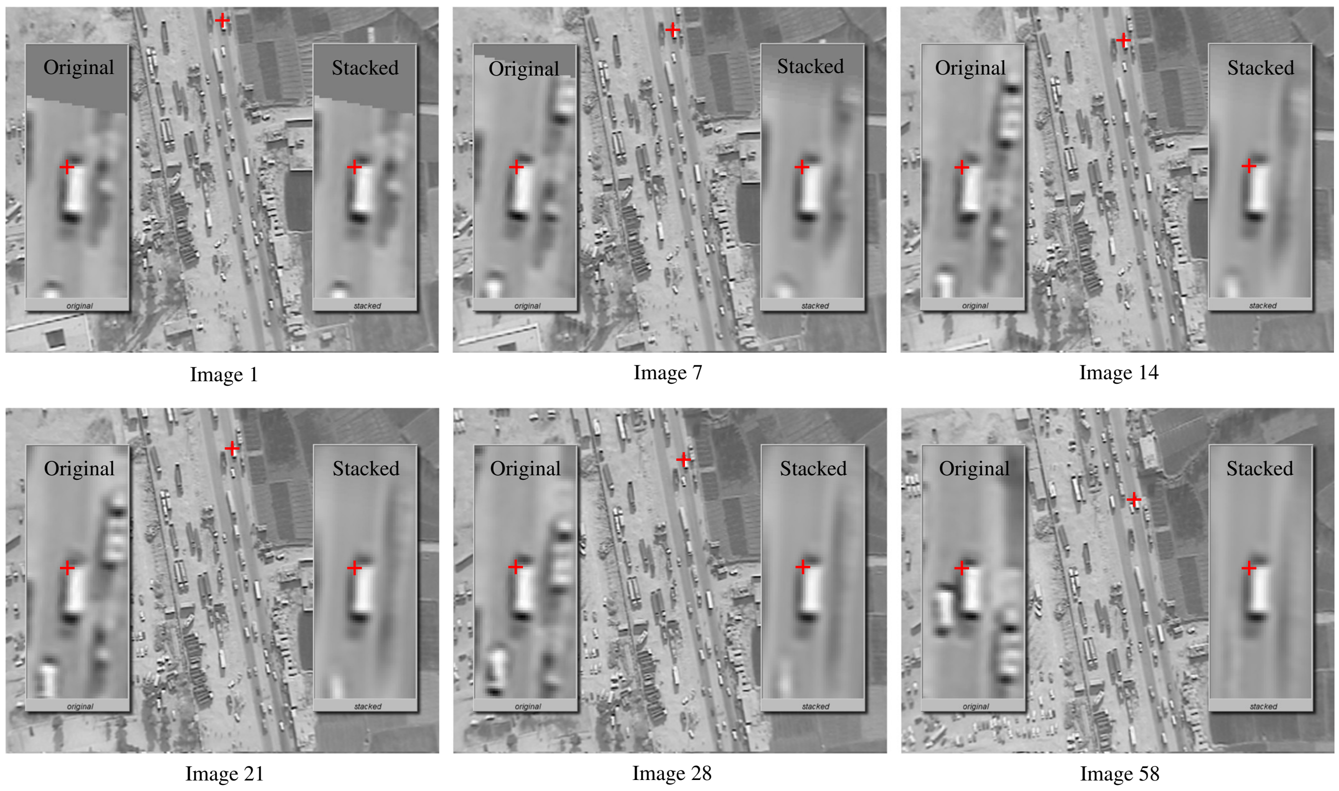


Fig. 1 Motivation for image stacking. The left part of each image shows the zoomed original area around the red cross rotated upright and the right part shows the stacked image. Please see the text for further description.

object detection and segmentation algorithms are implemented, improved by our image stacking approach, and evaluated using FMV sequences that contain occlusions, parked vehicles, and street textures.

The main contributions of this article are:

1. the introduction of image stacking for moving ground objects observed by a moving airborne camera to intentionally blur the surrounding stationary background,
2. handling and management of multiple potentially overlapping image stacks that occur when observing multiple moving ground objects in parallel, and
3. how to use the proposed image stacking approach to improve standard moving object detection and segmentation methods.

We briefly mentioned the proposed image stacking approach in prior work already,[15] but no details were given, such as how image stacks can be initialized, how multiple image stacks can be managed, and how our proposed approaches can be implemented efficiently. Furthermore, we provide an extensive study of the image stacking parameters that contribute most to the object detection and segmentation performance. Parts of this article are based on the first author's doctoral thesis.[16]

The remainder of this article is organized as follows: literature is reviewed in Sec. 2. The baseline moving object detection and segmentation algorithm are described in Sec. 3. The proposed image stacking approach is presented in Sec. 4. Experimental results are given in Sec. 5. We conclude in Sec. 6.

2 Related Work

Moving object detection in aerial videos is usually based on the detection of image regions that move independently of the camera followed by either segmentation techniques or machine learning approaches. Independent motion is then represented either by pixel clusters as done in background subtraction[8,17] and frame differencing[18,19] or by clusters of tracked local image features, such as Kanade–Lucas–Tomasi features or Harris corners.[20–22] Segmentation techniques can be based on thresholding,[23,24] morphological operations,[25] edge detection,[15,26] or superpixels[27,28] in combination with connected component labeling while machine learning approaches use trained classifiers in a sliding-window framework[29–31] often only applied to independently moving image regions.[32–34]

To further improve those methods, several approaches exist for spatial information fusion[15,26,31,35,36] and consideration of context knowledge, such as street networks or tracking statistics.[18,25,32,33,37–39] Temporal information fusion, however, is often introduced by using single or multiple object tracking that is based on initial detections. Pure feature tracking[20] is possible, too but not sufficient in general since no information is gained about individual object instances. Hence, it remains unclear if the track contains one object completely or partially, or even multiple objects moving in the same direction at similar velocity. Actually, our method is applied between object detection and object tracking and, thus, can be used to further improve multiple object tracking by providing more reliable initial detections.

Few approaches for image stacking exist in the literature that are somehow related to our proposed method. Prokaj and

Medioni[40] align and average several samples of a moving vehicle to set up a template for a regression tracker in wide-area motion imagery (WAMI) data. Ali et al.[41] calculated an intraclass variation between aligned samples of a vehicle and compared it to other vehicles in the scene (interclass variation) to reacquire the vehicle after an occlusion in FMV data. Finally, Mise and Breckon[42] apply multiframe super-resolution to low-resolution moving objects to enhance the object's image quality and, thus, improve the performance of correlation-based object tracking. However, while the above-mentioned approaches use image stacking to improve object tracking, our proposed image stacking approach is applied at an earlier stage to improve object detection and segmentation.

3 Moving Object Detection

In this section, we present the baseline processing chain, in which the proposed image stacking approach is embedded. An overview of the algorithm's pipeline is given in Sec. 3.1, and the two main submodules are described in more detail in Secs. 3.2 and 3.3.

3.1 *Concept*

The concept of the entire moving object detection process, including the proposed image stacking approach, is visualized in Fig. 2. Incoming image sequences are processed by two main modules: (1) independent motion detection and clustering and (2) object detection and segmentation. The first module is depicted in yellow color and mainly aims at determining motion that is independent of the camera motion. Image regions of independent motion are represented by motion clusters that can contain multiple objects or even parts of objects due to split and merge effects. There may even be no object at all, if the motion cluster is the outcome of imprecise image registration or parallax effects. In the second module that is depicted in red color, these motion clusters are used as regions of interest (ROIs) to detect and segment individual moving objects. Although "object segmentation" and "object detection" can be considered as two different topics in computer vision, they can be used to solve the same problem in the context of this article: improved localization and boundary determination of individual objects. The proposed image stacking approach (bright red color) is part of this module and supports the detection and segmentation process. It is described in detail in Sec. 4.

3.2 *Independent Motion Detection*

Independent motion detection is used to detect image regions that move independently of the camera. In our FMV data, these regions usually represent moving objects, such as vehicles or motorcycles. To detect independent motion, it is crucial to estimate the camera motion first. Therefore, we only use the images of the video sequence without any additional meta-information, such as camera calibration parameters, digital elevation models, or UAV flight parameters.

The image sequence is compensated for camera motion by the application of image registration.[1] Local image features are detected with subpixel accuracy and tracked over time.[43] We use Förstner–Harris corners[44,45] as local features. Corresponding corners between subsequent images are used to estimate frame-to-frame homographies as global image transformations to register images.[46] Outlier correspondences are removed by applying random sample consensus[47] with subsequent refinement. Using homographies is possible since the depth differences in the evaluated scenes are small compared to the distance of the observing camera, and the scene can therefore be approximated well with a ground plane. Those homographies represent the camera motion as shown in Fig. 2. Robust feature tracks that do not fit to the camera motion are assumed to be part of moving objects. Since these features have been tracked for a few frames, they can be represented not only by their positions but by motion vectors. Similar motion vectors are grouped to motion clusters. Therefore, single-linkage clustering[48] is performed based on position and velocity of the motion vectors. The choice of distance thresholds is based on the known GSD and the expected size of objects. As its output, the independent motion detection module delivers a set of motion vectors and motion clusters.

This process is visualized in Fig. 3. One original image taken from an image sequence of the test dataset is shown in Fig. 3(a). In Fig. 3(b), we see all detected and tracked local features represented by red vectors. There are usually around 20,000 motion vectors per image. We apply frame-to-frame homography estimation to calculate a global image registration that represents the camera motion. Using this homography,

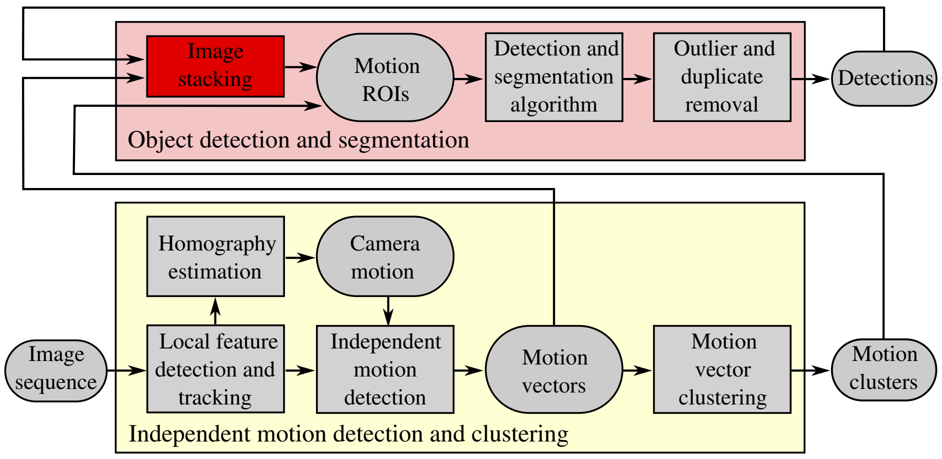


Image stacking
Motion ROIs
Detection and segmentation algorithm
Outlier and duplicate removal
Detections
Object detection and segmentation
Homography estimation
Camera motion
Image sequence
Local feature detection and tracking
Independent motion detection
Motion vectors
Motion vector clustering
Motion clusters
Independent motion detection and clustering

Fig. 2 The concept of the moving object detection and segmentation module. The proposed image stacking approach is highlighted in bright red color. Please refer to the text for further information.

we can distinguish between stationary and independently moving feature tracks. Stationary features are depicted in red color and motion vectors in yellow color in Fig. 3(c). Clustered motion vectors are used to calculate a local image registration for local image alignment. On average, each moving object in the test dataset produces a cluster of 15.57 motion vectors. Hence, robust image alignment based on small moving objects, which is crucial for the proposed image stacking approach, is possible as we need at least two motion vectors to estimate the stack alignment (translation and rotation). Motion vector clustering leads to the cyan boxes in Fig. 3(d), where each box represents one motion cluster.

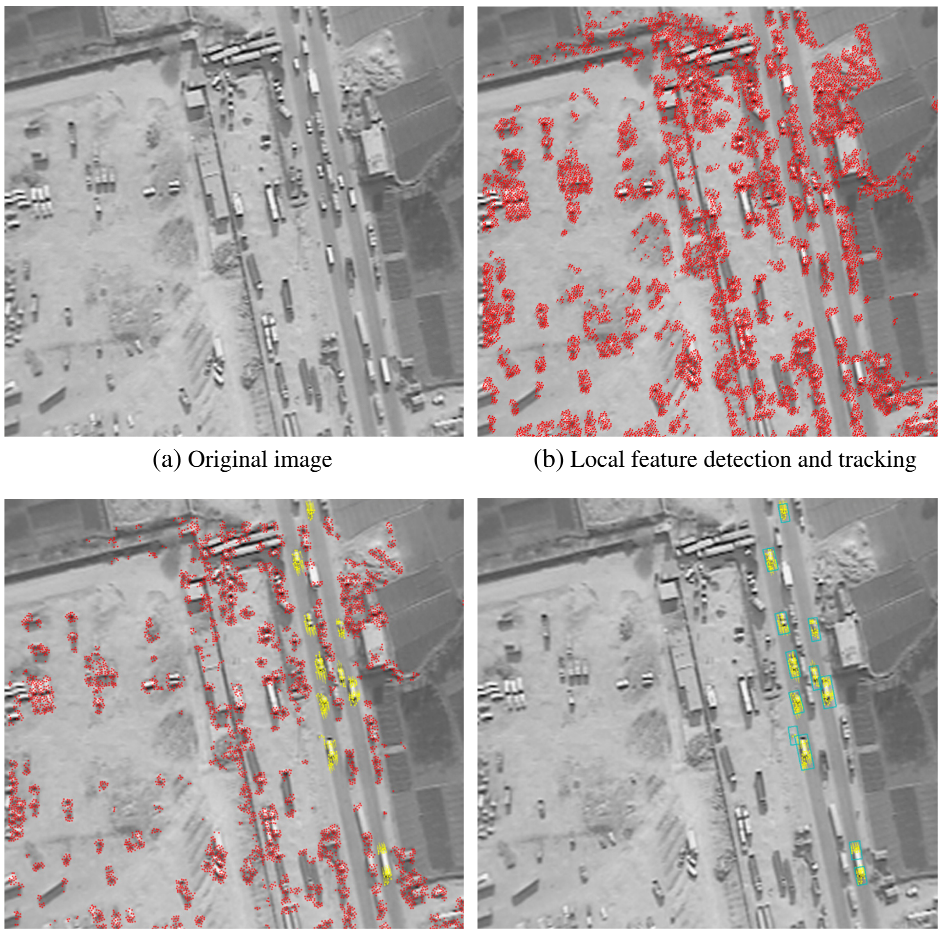
Instead of motion vector clustering, background subtraction[8,17] or frame differencing[18,19] is the popular method for independent motion detection. However, for image stacking, we want to use the temporal information that is implicitly contained in the motion vectors. Furthermore, motion vector clustering is less prone to be affected by noise.[20,22,49]

3.3 *Object Detection and Segmentation*

For object detection and segmentation, we use two different approaches and implement them as a baseline that we want to improve by the application of image stacking. The first approach is motivated by appearance-based object detection using machine learning[34] and the second approach is based on object segmentation by the clustering of edge pixels.[49] The result of both methods is bounding boxes that surround the moving objects. For simplicity and to be consistent with existing literature, we call those bounding boxes "detections" even if they are the result of object segmentation.

The feature clustering during independent motion detection that provides the motion clusters is prone to produce split and merged detections of moving objects. Split detections can occur for weakly textured vehicles, and merged detections appear due to the similar motion direction and velocity in dense urban traffic.[49] Thus, the motion clusters are considered as initial object hypotheses and define a search space. Before object detection and segmentation are applied, the motion clusters are spatially extended in the motion direction to fully contain objects that are detected partially by independent motion detection. Those extended regions are denoted by "motion ROIs."

For object detection, we train a classifier using 790 vehicle samples and 790 nonvehicle samples. Each sample has a size of 16×32 pixels. We use Integral Channel Features as a descriptor and train a boosted decision trees (BDT) classifier using Real AdaBoost.[50] This classifier is


(a) Original image (b) Local feature detection and tracking

(c) Independent motion detection (d) Motion vector clustering

Fig. 3 Example for independent motion detection and clustering. Stationary local features are visualized in red color, and motion vectors moving independently of the camera motion are depicted in yellow. Motion clusters are represented by cyan boxes.

applied locally to the motion ROIs in a sliding-window framework. To consider different vehicle sizes, image rescaling is done with three different scales.[34] Note that, in this work, image rescaling is not used to detect objects in different distances (as done in pedestrian detection, for example), since we perform a scale normalization using the known GSD before we run the detector. Finally, a nonmaximum suppression is applied based on the intersection-over-union (IoU) criterion and the classifier confidence.

For object segmentation, we calculate gradient magnitudes using local binary patterns[16] and use quantile-based thresholding for binarization.[51] A typical value for the quantile q is 0.15. Thus, we assume that 15% of the pixels inside a motion ROI belong to moving objects. After the application of morphological closing and connected-component labeling, we calculate bounding boxes for each blob that exceeds a certain area or size (w.r.t. number of pixels). Each bounding box represents one moving object. The above-described approach is chosen because it is able to outperform[16] other segmentation methods that are based on global probability of boundary,[52] superpixels,[53] spatiotemporal saliency,[54] top-hat transform,[25] or Canny gradient thresholding.[26] The main reason, therefore, is the image quality, which is severely limited by low resolution and atmospheric effects causing noise and blurry object edges.

Finally, duplicate and outlier detections are removed. Duplicate detections occur due to overlapping motion ROIs. They can be rejected by applying the earlier mentioned nonmaximum suppression for object detection and by fusing all overlapping bounding boxes that exceed a certain IoU threshold for object segmentation.[16] Outliers are detections that no motion vectors can be associated with. Therefore, we simply check the number of motion vectors inside the detection bounding box. Detections with <4 associated motion vectors are rejected, as we assume them to be stationary.[16,34]

4 Image Stacking for Moving Objects

The main contribution of this article is presented in this section: (1) the introduction of image stacking for moving objects, (2) handling and management of multiple potentially overlapping image stacks, and (3) the replacement of motion clusters by image stacks to improve object detection and segmentation.

Object detection and segmentation as presented before are working well as long as the object appearance is clearly visible. However, there are three effects that can affect the object appearance and decrease the detection and segmentation performance by causing inaccurate object boundaries or even producing false positive (FP) or false negative (FN) detections:

- partial occlusions by trees, power supply lines, or buildings,
- stationary objects close to the observed object, such as parked vehicles or buildings, and
- street textures, such as cobblestones or road markings.

To suppress these effects, image stacking is introduced to mask and blur the stationary background around a moving object. Several motion ROIs of the same object in subsequent images are registered and aligned using the motion vectors, i.e., we apply local image registration as described and visualized in Figs. 3(c) and 3(d). Each aligned motion ROI is one layer of the image stack. By calculating the pixelwise mean or median pixel value, the stationary background and even other moving objects with a nonzero relative velocity compared to the observed object are blurred as seen in Fig. 1. The red cross visualizes a motion vector that is tracked for 250 consecutive images. The image region ("stack region") where stacking is applied is generated using the motion direction and fixed values for width and length of the stack area. This stack region is zoomed and visualized without stacking in the left and with stacking in the right area of each image. The center of the stack region is fixed by the position of the motion vector (red cross). After that, the stack region, which is oriented in the motion vector direction in the original image, is rotated upright and added to the stack. The pixelwise mean image of the entire stack is visualized in Fig. 1. For a pixel position (x, y) in the stack S and in each aligned image I_h with $h \in \{1, \dots, H\}$, the corresponding pixel value $S(x, y)$ is given as

$$S(x, y) = \frac{\sum_{h=1}^{H} I_h(x, y)}{H}, \tag{1}$$

where H is the total number of stack regions added to the image stack and is called the "stack height." Since the motion direction of single motion vectors fluctuates slightly, the direction of each motion vector can be smoothed by using the direction of the related motion cluster. This is important to avoid blurring of the edges of the observed object. After 28 images, the stationary background is fully masked while the overtaking vehicle disappears later, after 58 images, due to its smaller relative velocity compared to the observed object. Parked vehicles or buildings can be very close to the observed object as seen in image 28, but image stacking successfully masks these stationary objects.

4.1 *Concept*

The concept of the proposed image stacking approach is visualized in Fig. 4. It shows the processing that is implemented in the image stacking module depicted in bright red color in Fig. 2. As there can be multiple moving objects in the scene, there usually are multiple image stacks and we aim at having at least one image stack per object. New image stacks are initialized either by using motion vectors or detections. New motion vectors are associated to existing stacks based on position and motion to improve the robustness of image alignment for small moving objects. Image stacks are updated by adding the corresponding aligned image area of the current image to the stack. Finally, image stacks replace related motion ROIs in the original unstacked image for the application of the object detection and segmentation algorithms. As seen in Fig. 2, image stacking can be skipped if image alignment is not robust or if the stack height H is not sufficiently large, yet. In this case, the motion clusters in the original unstacked image are used for object detection and segmentation.

4.2 *Image Stack Initialization*

The implementation of image stacking in a fast and efficient way is not easy. In a feasibility study,[15] it was demonstrated that generating and initializing a stack for each motion vector in the image can lead to hundreds of detections (and stacks)

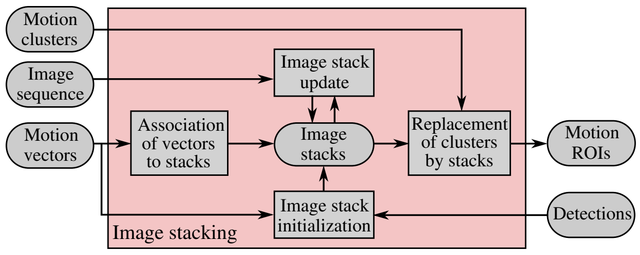


Fig. 4 The concept of the image stacking submodule. Please refer to the text for further information.

per object, but adequate duplicate removal of sufficiently overlapping bounding boxes improves the performance of object segmentation with respect to detection rate and precision. However, this method is highly inefficient w.r.t. processing time and needs to be improved for implementation. The main difference between the feasibility study and this concept is the introduction of "master vectors" as representatives for clustered motion vectors. Each master vector is a virtual motion vector and owns exactly one image stack. In Fig. 5, master vectors are visualized as colored crosses. Thus, the total number of image stacks decreases from several hundred to <50 per image. The master vector lies in the center of its related image stack.

To initialize an image stack, one has to detect a cluster of similar motion vectors. This cluster can be different from the motion clusters, which are the result of independent motion detection and clustering. The reason is that moving objects driving with a small relative velocity close to each other may be clustered together in the same motion cluster while image stacking aims at initializing at least one stack per object. Only in this way, it is possible to benefit from the blurring effect since the first object is assumed to be focused and sharp in the first stack while the second object gets blurred over time due to its velocity difference, and vice versa, the second object stays sharp in the second stack while the first object is blurred. If only one stack is initialized for two objects, the blurred object may be missed by the detection

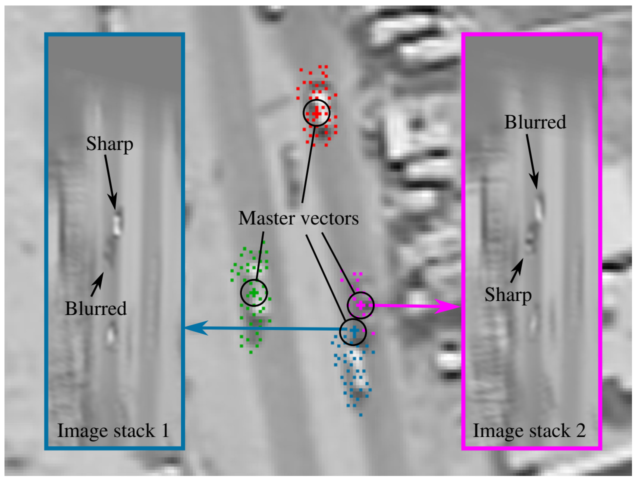


Fig. 5 Image stacks for two vehicles with small relative velocity of about 10 km/h (about 0.4 pixels per frame). The master vector of each stack is visualized as a colored cross. The first vehicle is sharp and the second is blurred in the first stack, while the first vehicle is blurred and the second is sharp in the second stack.

and segmentation algorithm. This effect is visualized in Fig. 5 with two vehicles overtaking each other. Each master vector has between 3 and 50 related motion vectors. These related motion vectors are represented by dots in the same color as their master vector. The blue master vector is not in the center of its cluster since it was initialized while both objects were merged in one large bounding box. In contrast, the magenta master vector was initialized later when both objects were detected separately. After 1 s or 25 stacked images, the pixelwise mean image shows that usually only one object stays sharp per stack. Two strategies have been implemented for image stack initialization.

1. The results of object detection and segmentation are called "detections." These detections are assumed to have a higher detection rate compared to the motion clusters that are prone to produce splits and merges. By using them as feedback information to initialize stacks, the ratio of stacks to objects is close to 1, which is most efficient. Even for merged detections there will be separate image stacks, if each single object has been correctly detected at least once. Immediately after that, the individual stacks are initialized. This is the case in Fig. 5. The blue master vector is not in the center of its cluster as it was initialized while both objects were merged in one large bounding box. The magenta master vector was initialized later when both objects were detected separately for at least one time step. To avoid any information loss, the blue master vector is kept and not reinitialized. However, objects that cannot be detected and segmented separately at all end up in a common image stack. In such a case, the blurring effect is assumed to be weak for each object, and detection performance is similar to using no image stacks.
2. A weak detection performance can lead to the initialization of image stacks for merged detections. In this way, image stacking may even decrease the overall detection performance. To use the motion vectors and motion clusters instead of the detections, "k-means clustering"[55] is introduced. k is the number of clusters after clustering and can be chosen based on the known standard vehicle size and the size of the related motion cluster. Then, the motion vectors are grouped using position and motion. This approach is less efficient since usually more stacks than objects are initialized and the calculation of k-means is time-consuming.

Both approaches are evaluated in Sec. 5.

4.3 *Association of Motion Vectors to Image Stacks*

Due to occlusions or varying object appearance, existing motion vectors can disappear and new motion vectors can appear at any time in the image sequence. A master vector is deleted if it loses all related motion vectors. While each master vector can have many related motion vectors, each motion vector can be related to only one master vector. A new motion vector is associated to a master vector by applying a k-nearest neighbors (k-NN) algorithm.[48] k-NN is a voting algorithm searching for the k nearest motion vectors in the image area around the new motion vector. Each neighbor votes for its master vector, and the new motion vector is

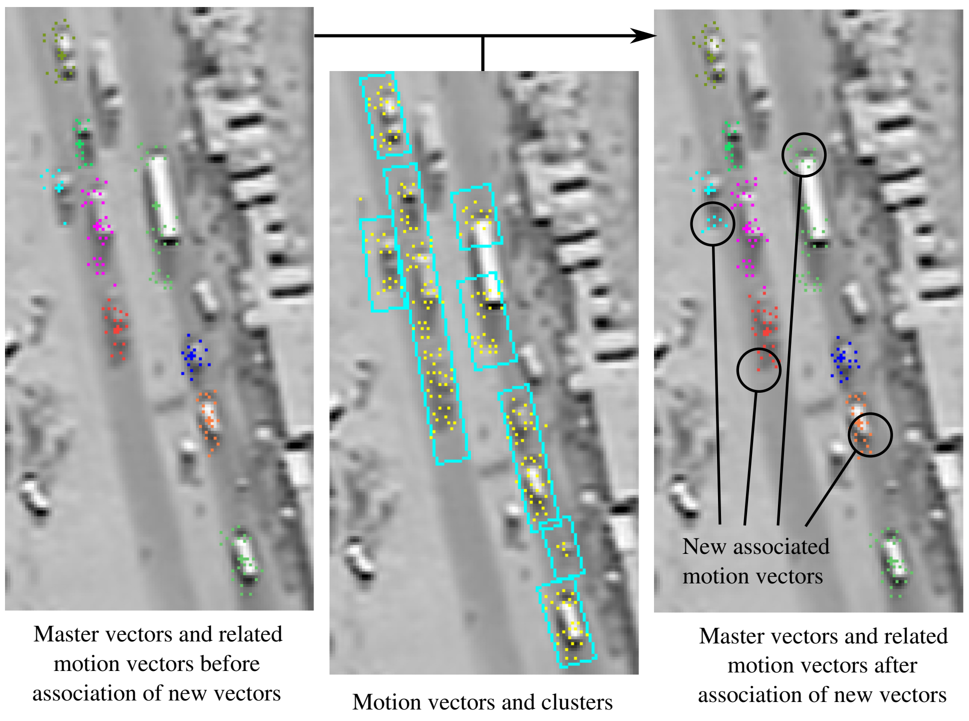

Master vectors and related motion vectors before association of new vectors
Motion vectors and clusters
New associated motion vectors
Master vectors and related motion vectors after association of new vectors

Fig. 6 Association of new motion vectors to master vectors (colored crosses). Associated motion vectors are visualized as dots in the color of its master vector.

associated to the master vector with the most votes. The search space can be significantly reduced by only considering motion vectors in the same motion cluster. A typical value for k is 3. This process is shown in Fig. 6. Each master vector (image stack) is visualized with a cross in a different color. The associated motion vectors are displayed as dots in the same color as their related master vectors. After k-NN using the motion vectors and clusters, the new motion vectors are associated to master vectors.

4.4 *Image Stack Update*

With each new incoming image, image stacks are updated in two steps: (1) the position of the master vector is updated and (2) the current stack area is added as a new layer to the image stack. Since the master vector is a virtual motion vector, its position and orientation can only be updated by its related motion vectors. Each related motion vector stores its relative position and orientation to the master vector right after it has been associated. This way, it can suggest a position and orientation of its related master vector in the current image. The final master vector position and orientation are determined by calculating the median of all suggested positions and orientations. Hence, even few incorrectly tracked related motion vectors will not affect the final master vector.

The stack area surrounding the updated master vector is rotated in upright position and added to the image stack as shown in Fig. 7. There are two considered strategies to arrange the image stack.

4.4.1 *Accumulation image*

One image with the size of the rotated stack area is initialized. Each pixel value is zero. To append a new stack area, it is rotated in an upright position, and each pixel value of the rotated stack area is added to the accumulation image. The stack height is stored by incrementing a counter variable with each appended stack area. The accumulation image is a very efficient way to arrange the image stack since only one image has to be kept in the memory and new layers are simply added to this image. The motion ROI can be calculated very fast by dividing each pixel by the stack height. This is the mean pixel value of all stack areas appended to the stack just as described by Eq. (1). The main disadvantage of this approach is that a slightly varying object appearance due to changes in the camera angle or UAV altitude will strongly affect the resulting motion ROI by blurring the observed object.

4.4.2 *Circular buffer*

To avoid this self-blurring effect, old stack areas can be replaced after a certain time and, hence, the stack can be arranged as a circular buffer. The size of the circular buffer directly corresponds to the stack height. It is fixed and determined *a priori*. In this way, the maximum number of stack areas in the stack is limited. New stack areas are added to the buffer and as soon as the buffer is full, the oldest stack area is replaced by the new one. The motion ROI is calculated either by the pixelwise mean or median pixel value of all stack areas. This is much more time-consuming than the first approach with the accumulation image. Figure 7 shows a circular buffer with stack height $H = 5$ and where the stack area at position 4 is replaced. The standard buffer size used in this article is $H = 50$ since all stationary objects and most objects moving relative to the observed object disappear in the motion ROI.

The comparison and evaluation of both approaches will be presented in Sec. 5. An image stack is initialized as soon as the master vector is initialized. Hence, a region of

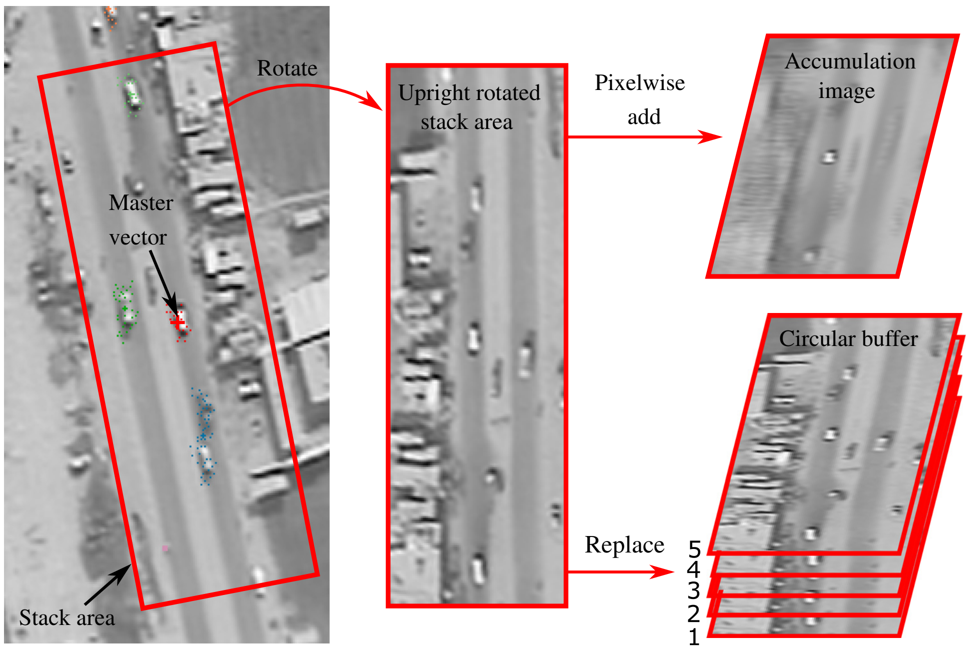


Fig. 7 Image stack update and stack arrangement either as accumulation image or circular buffer.

the stack area can be outside of the image borders. To add the whole area to the stack but minimize the influence of the outer region for the calculation of motion ROIs, the pixels in these regions are set to the mean of the gray-value range. We set this value to 127.

4.5 *Replacement of Motion Clusters by Image Stacks*

The replacement of motion clusters by motion ROIs that are provided by image stacking is optional. Motion ROIs are used, if suitable stacks are available, and motion clusters are used otherwise. A suitable image stack has to meet the following four criteria:

1. The velocity and the direction of the master vector have to be close to the velocity and the direction of the motion cluster.
2. The extended motion cluster has to be fully inside the stack area.
3. The master vector has to be inside the motion cluster.
4. The stack height has to exceed a minimum threshold $H_{\min}$.

This process of finding suitable image stacks is shown in Fig. 8. Image stacks shall be found for the lower motion cluster (cyan rectangle). The extended motion cluster is visualized with a dashed cyan rectangle. There are two vehicles inside the motion cluster, and for each vehicle, an image stack has been initialized (shown in magenta and blue color). For both stacks, all four criteria are fulfilled, so two suitable image stacks have been found. These image stacks replace the original motion cluster by replacing the original image with the stacked image. This stacked image can be calculated using the pixelwise mean of all images in the accumulation image or all images in the circular buffer as described in Eq. (1) or the pixelwise median of all images in the circular buffer using the following equation:

$$S(x,y) = \operatorname*{median}_{h \in H}[I_h(x,y)], \tag{2}$$

where S is the stacked image, I_h is the h'th original image in the stack at pixel position (x,y), and H is the size of the circular buffer with $h \in \{1,\dots,H\}$.

4.6 *Discussion*

Image stacking is introduced to improve the object detection performance by considering temporal context. Since only motion vectors and no object representations are used, image stacking takes place on feature level before multiple object tracking is applied on object level. Object detection and segmentation are applied to each motion ROI (stacked image) separately and the original image motion cluster is discarded, if suitable image stacks have been found. As soon as a motion cluster has been replaced by suitable image stacks, the aim is to detect only the sharp object in each stack and, thus, avoid a potential merged detection containing background structures or several moving objects. If a blurred object is detected, too, this detection is suppressed by the outlier and duplicate removal module. In contrast to the first implementation of image stacking,[15] the approach presented here can be processed in real time. The runtime will be analyzed in detail in Sec. 5.

Figure 9 shows examples for each of the three situations where image stacking improves object detection as mentioned in the beginning of Sec. 4. These are (1) partial occlusion by a tree in Fig. 9(a), (2) a parked vehicle close to the observed moving vehicle in Figs. 9(b) and 9(c), and (3) disturbing street textures in Fig. 9(d). The observed object is located in the center of each image. Since image stacking will improve object detection and segmentation only in such situations, only minor improvement of detection performance is expected. Furthermore, image stacking can even cause additional false detections if moving objects with small relative motion merge due to blurring. This effect becomes apparent in Fig. 5.

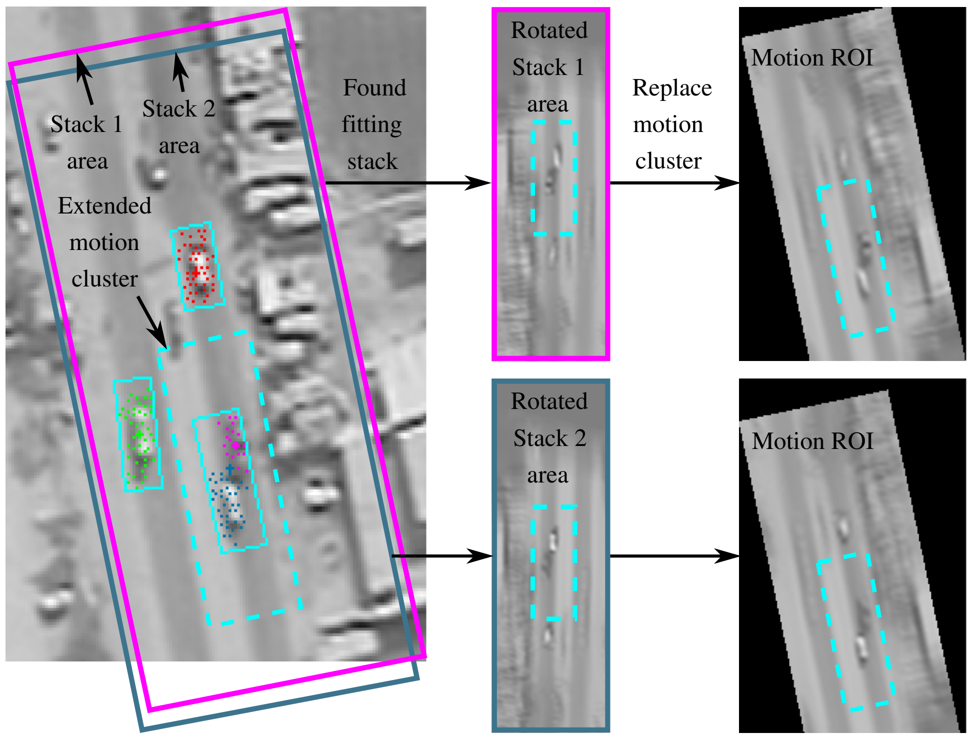


Fig. 8 Replacement of motion clusters by image stacks (motion ROIs). Two suitable image stacks are found for the extended motion cluster (dashed cyan rectangle) and replace the original image of the motion cluster.

Important parameters for image stacking are the size of the stack area A, the minimum stack height H_{min}, and the stacking strategy. A has to be chosen large enough so that the stack region fully covers even large objects such as trucks. H_{min} is the threshold of stack height H that has to be exceeded before a stack is returned as motion ROI. This is important since small stacks of $H = 10$ or less are prone to cause objects merging due to blurring. If H_{min} is too large (e.g., 50 or 100), fewer image stacks are used, and potential benefit decreases as objects may already be outside of the camera view before H_{min} is reached and the motion ROI is returned. Good results have been achieved with $A = 100 \times 300$ pixels, $H_{\text{min}} = 25$, and circular buffer instead of accumulation image. The influence of varying these parameters for object detection and segmentation is analyzed in Sec. 5.

5 Experiments and Results

Four gray-value video sequences are used in our experiments. Sample frames for each sequence are visualized in Fig. 10. While the first three sequences SEQ 1, SEQ 2, and SEQ 3 are coming from your own top-view aerial video data, the fourth is the EgTest01 sequence taken from the publicly available video verification of identity (VIVID) dataset.[56] To the best of our knowledge, no further publicly available FMV datasets currently exist that contain challenging urban

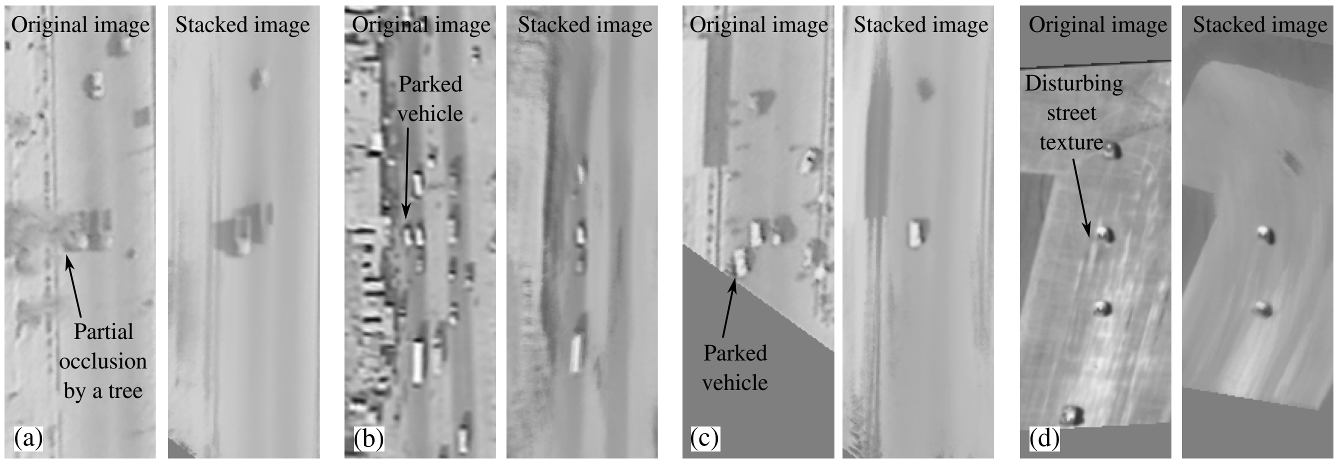


Fig. 9 Examples for successful application of image stacking. The observed object is located in the center of each image.

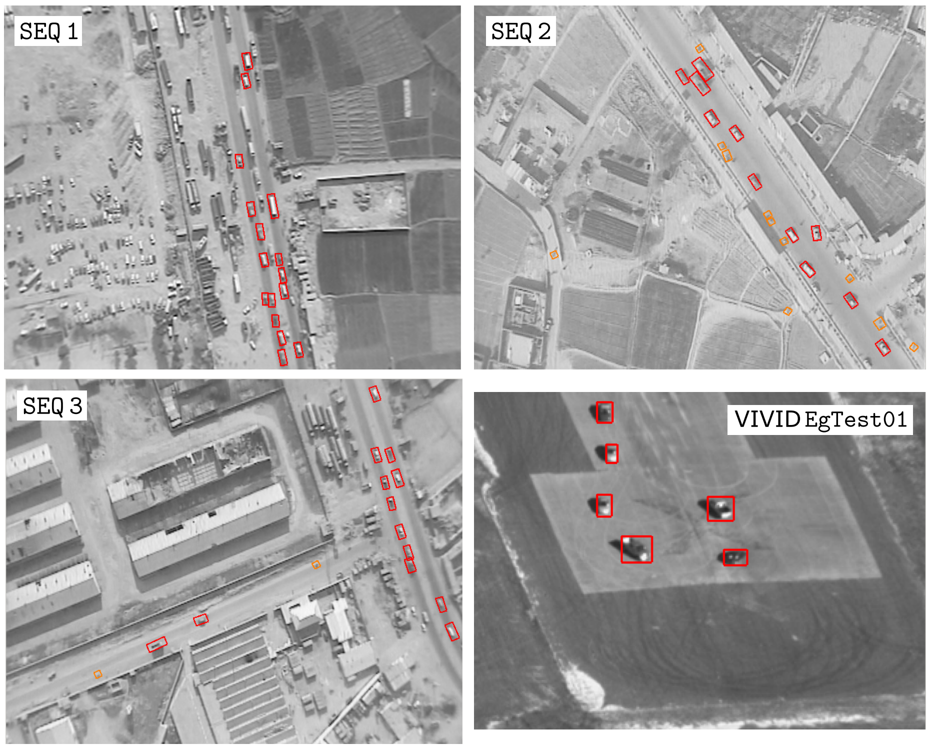


Fig. 10 Aerial videos used for the experiments. Moving objects GT is represented by bounding boxes: red color for vehicles and orange color for motorcycles, bicycles, or pedestrians. Sequences 1–3 are taken from our data and the rightmost one from the VIVID dataset.[56]

scenarios with dense traffic. WAMI datasets, such as Columbus large image format[57] or Wright-Patterson air force base,[58] cannot be used here due to their low frame rate.

Although the EgTest01 sequence is recorded in oblique view, we consider this sequence in our experiments as we want to show some limitations of applying our proposed method to oblique view videos. The number of frames is approximately between 200 and 2000, and the frame rate is between 25 and 30 Hz. Our own data consist of busy urban scenes. There are between 5 to 20 moving objects per frame. With a GSD of about 0.345 m/pixel, a standard car covers an area of $\sim 10 \times 20$ pixels in the image. Some statistics of the evaluated sequences are given in Table 1.

Moving objects are represented by bounding boxes. The ground truth (GT) was labeled manually. We distinguish between vehicles (red boxes in Fig. 10) and nonvehicles, such as moving motorcycles or pedestrians (orange boxes in Fig. 10). This is important as we compare a model-based vehicle detection approach and a model-free object segmentation approach. Since the vehicle detection approach is not able to detect nonvehicles, we declare nonvehicles as “do not care” objects for the quantitative evaluation to guarantee a fair comparison between the two approaches. In the original GT of the EgTest01 sequence, only one object is labeled for an evaluation of single object tracking. We extended the GT manually to all six moving vehicles.

A detection is true positive (TP), if the bounding boxes of GT and detection overlap at least 10% w.r.t. to the IoU criterion.[59] In contrast to the PASCAL criterion of 50% overlap,[60] we choose this small overlap as for small objects covering only 50 to 200 pixels even small deviations in size or position of the detection bounding box from the GT bounding box can induce a significantly lower overlap. For the quantitative evaluation, we use standard measures, such as precision, recall, and f-score,[61] as well as the normalized multiple objects detection accuracy (N-MODA) and the normalized multiple object detection precision (N-MODP).[62]

5.1 *Parameter Optimization*

Four parameters are selected for optimization: minimum stack height $H_{\min}$ and stack area A are continuous parameters, whereas stack initialization and stack arrangement are discrete parameters. The height of a stack has to exceed $H_{\min}$ before the motion cluster is replaced by the stack for object detection and segmentation. If $H_{\min}$ is not exceeded, the motion cluster is used instead. A is the spatial extend of a stacked area in the image. With a larger size, it is more likely that extended motion clusters are inside the stack area and can be replaced by stacks. Two approaches for stack initialization are proposed in Sec. 4.2: initialization by k-means clustering of moving corners and initialization by detections. Finally, two different methods are introduced for stack arrangement in Sec. 4.4: accumulation image and circular buffer. The stack image of a circular buffer can be calculated by pixelwise mean or median. Maximization of the f-score for SEQ 1 is chosen as optimization criterion.

The optimization results are visualized in Fig. 11. Stack initialization is evaluated for both the model-based vehicles detection approach and the model-free object segmentation approach in Fig. 11(a). For both approaches, k-means clustering performs worse compared to initialization by detections, so the latter one is chosen to initialize stacks. In Fig. 11(b), the f-score of $H_{\min}$ for all three stack arrangement methods is depicted. $10 \le H_{\min} \le 250$ is the chosen range clearly demonstrating that $H_{\min} = 25$ is the best value for

Table 1 Statistics of the aerial video datasets.

			GT			
Video	Frames	Frame rate (Hz)	Moving objects	Single detections	Moving vehicles	Single detections
SEQ 1	400	25	42	4785	40	4731
SEQ 2	200	25	39	2662	18	1373
SEQ 3	400	25	38	5023	36	4946
EgTest01	1821	30	6	6866	6	6866

each method. Figure 11(c) shows the optimization for stack area A. The f-score of the three stack arrangement approaches is plotted against the stack area size in pixels. With 1∶2 and 1∶3, two different ratios of width to length of A are evaluated. The variance of the f-score is smaller for ratio 1∶3 compared to 1∶2, so 1∶3 is further considered. 115×345 pixels is the best value for the stack area since there is no improvement of the f-score anymore for larger values. In all evaluations, the accumulation image achieves a worse maximum f-score than the circular buffer. Eventually, the circular buffer with pixelwise median image is chosen since it performs slightly better compared to the pixelwise mean image.

5.2 *Experimental Results*

In this section, the object segmentation and the vehicle detection approach are evaluated with and without the proposed image stacking method for each of the four aerial videos. The aim is to determine the situations in which the application of image stacking is beneficial and to measure the potential improvement. The quantitative evaluation is presented in Table 2. Bold font indicates an improvement compared to the approach without image stacking. In SEQ 1, image stacking reduces the number of FPs and FNs by 76 for the segmentation approach and by 14 for the detection approach, respectively. A similar result is achieved for SEQ 2, where the number of FPs and FNs is decreased by 29 and 19, respectively. The improvement of the f-score is between 0.002 and 0.011.

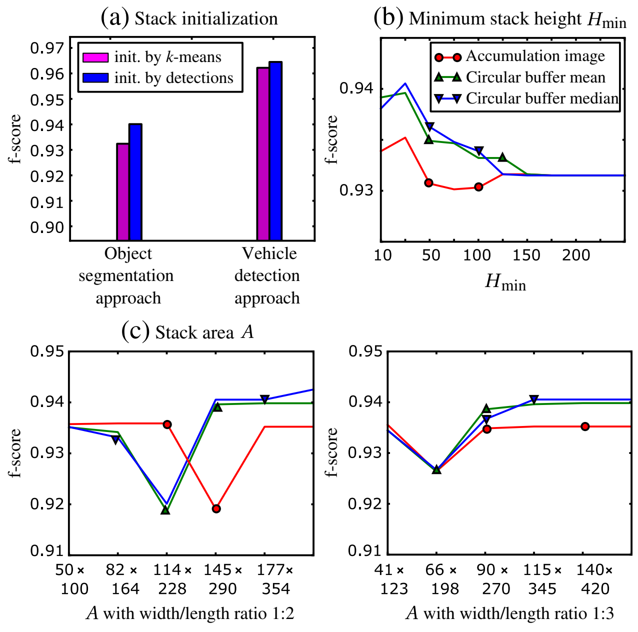


Fig. 11 Optimization of the stacking parameters using f-score and SEQ 1.

However, image stacking does not really improve the results for the sequences SEQ 3 and EgTest01. The number of FPs and FNs is reduced by only 16 and 15 for object segmentation, respectively. At the same time, the performance of the vehicle detection approach is even decreased. In SEQ 3, there are no occlusions and only few merged detections, so that image stacking cannot generate much benefit here. Furthermore, slightly imprecise tracking of the motion vectors can result in blurry observed objects, which lead to an even higher number of FN detections. The reason is that due to object blurriness, the quantile-based threshold for object segmentation and the classifier's confidence value threshold for object detection are not exceeded anymore. In the EgTest01 sequence, there are no merged detections, no partial occlusions, and only few disturbing street textures. Image stacking works well as long as the object appearance is stable. If this is not the case, the object gets blurred or deformed in the image stack. As mentioned earlier in Sec. 5, the camera angle and, thus, the vehicle appearance are varying in EgTest01. This is difficult to handle for the classifier that was only trained with top-view samples. It is even more challenging, when the turning vehicles at the beginning of the scene get deformed during image stacking as demonstrated in Fig. 12.

In general, there is stronger improvement in both accuracy (f-score and N-MODA) and precision (N-MODP) for object segmentation compared to the vehicle detection approach. The main reason is that outlier removal as described in Sec. 3.3 is able to cover most situations where stacking can improve vehicle detection: due to the fixed size of the sliding window, usually no undersegmentation occurs in case of merged detections. Instead, objects are either missed or FP detections appear. These FPs are reduced by both image stacking and outlier removal. At the same time, undersegmentation as occurring regularly in object segmentation cannot be handled well by outlier removal but only by image stacking.

The qualitative evaluation is done separately for object segmentation in Fig. 13 and vehicle detection in Fig. 14. Four image sections are chosen for each of the two methods. GT is visualized in the top row, and the approach without and with image stacking is depicted in the center and lower row. Orange GT bounding boxes represent moving motorcycles,

Table 2 Quantitative evaluation for image stacking.

Video	Evaluation measure	Object segmentation	Object segmentation + stacking	Vehicle detection	Vehicle detection + stacking
SEQ 1	TP	4322	**4389**	4463	4461
	FP	223	**214**	83	**67**
	FN	409	**342**	268	270
	Precision	0.952	**0.954**	0.982	**0.985**
	Recall	0.913	**0.928**	0.943	0.943
	f-score	0.931	**0.940**	0.962	**0.964**
	N-MODA	0.866	**0.882**	0.925	**0.928**
	N-MODP	0.621	**0.632**	0.696	0.694
SEQ 2	TP	1265	**1274**	1181	**1196**
	FP	190	**170**	47	**43**
	FN	108	**99**	192	**177**
	Precision	0.869	**0.882**	0.961	**0.965**
	Recall	0.921	**0.928**	0.860	**0.871**
	f-score	0.894	**0.905**	0.908	**0.916**
	N-MODA	0.782	**0.804**	0.825	**0.839**
	N-MODP	0.573	**0.575**	0.593	0.593
SEQ 3	TP	4755	4752	4802	4761
	FP	200	**181**	165	**140**
	FN	191	194	144	185
	Precision	0.960	**0.963**	0.967	**0.971**
	Recall	0.961	0.961	0.971	0.963
	f-score	0.961	**0.962**	0.969	0.967
	N-MODA	0.920	**0.924**	0.937	0.934
	N-MODP	0.608	**0.610**	0.705	0.699
EgTest01	TP	6812	6807	6726	6677
	FP	128	**108**	73	**69**
	FN	54	59	140	189
	Precision	0.982	**0.984**	0.989	**0.990**
	Recall	0.992	0.991	0.980	0.972
	f-score	0.987	**0.988**	0.984	0.981
	N-MODA	0.973	**0.976**	0.968	0.962
	N-MODP	0.527	**0.544**	0.526	0.518

Note: Improvement by image stacking is highlighted in bold.

bicycles, or persons that are not considered for evaluation (ignore regions). The cyan and red boxes in the center and lower row visualize motion clusters and detections, respectively. The red arrows point at the improvement by image stacking. In Figs. 13(a) and 13(b), typical merge situations are shown where two objects drive close to each other and cause undersegmentation. They can be separated by image stacking since each object has its own stack without

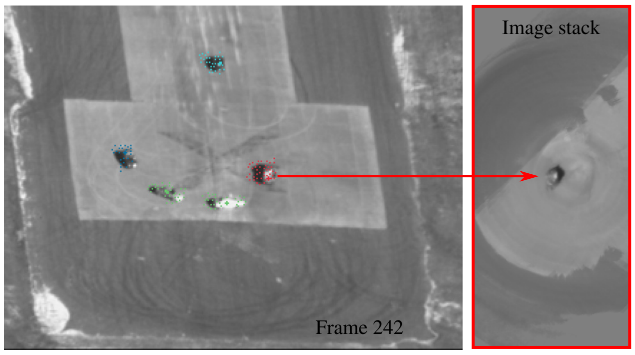


Fig. 12 Image stacking may not work for turning vehicles and oblique camera angle (EgTest01). The stacked object is blurred and its shape is deformed. This can cause additional FNs for the vehicle detection approach.

disturbing background. The right truck in Fig. 13(c) is partially occluded by a tree, while there is disturbing street texture in Fig. 13(d). Object segmentation is still possible but imprecise. Image stacking compensates for the occlusion and the street texture leading to more precise segmentation results. In Fig. 14, merged detections (a), FPs (b), and FNs (c) occur for the vehicle detection approach. The reason is ambiguities of the classifier when there are more prominent object contours between objects than for the real individual objects. In these cases, image stacking helps since the classifier confidence is higher inside the individual stacks with exactly one sharp object per stack. The partial occlusion in Fig. 14(d) is successfully handled with image stacking although only the shadow is detected for the left truck.

In summary, image stacking is able to improve both the object segmentation and the vehicle detection approach. The improvement may look minor; however, there are only few situations where the application of image stacking is beneficial, such as partial occlusions, merged detections, and ambiguous detections in dense traffic. There is less improvement for vehicle detection since both image stacking and outlier removal handle similar problems and, thus, complement each other. In our data, vehicle detection outperformed object segmentation in all sequences. The reason is that vehicle detection, which is based on a sliding-window approach, makes assumptions about vehicle size and appearance that are satisfied in the evaluated sequences. The model-free object segmentation approach achieves better performance in cases where those assumptions are violated, such as in sequence EgTest01.

The runtime for managing about 20 stacks per image varies between 48 and 870 ms. This strong variation is dependent on the choice of the stack arrangement: while stack initialization and association of motion vectors to stacks (Sec. 4.2), and replacement of motion clusters by stacks (Sec. 4.3) take less than 20 ms altogether, stack update

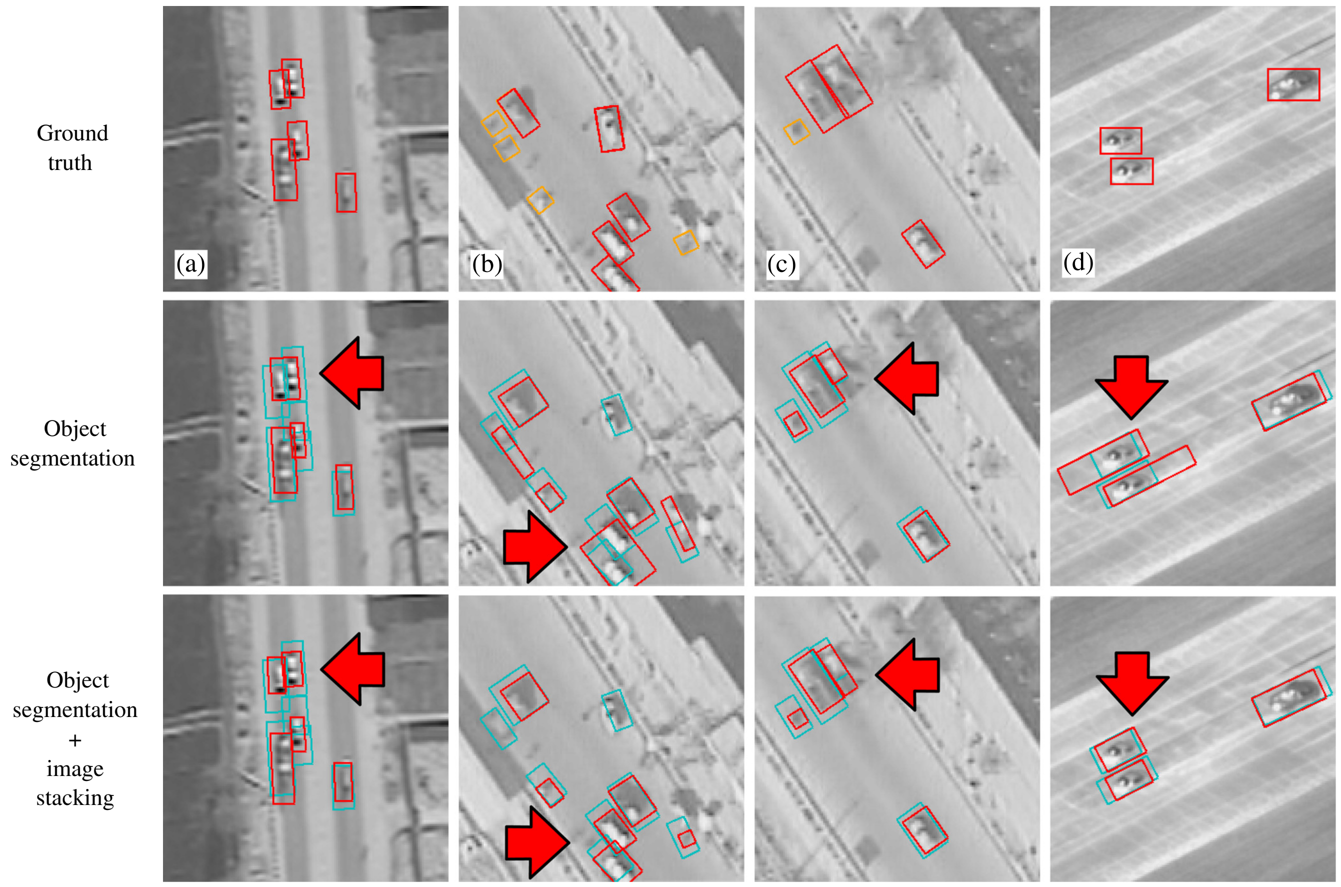


Fig. 13 Examples taken from the qualitative evaluation of the proposed image stacking approach applied to moving object segmentation. Merged detections can be handled, if there is a relative velocity difference between the vehicles (a) and (b). Furthermore, imprecise segmentation due to occlusion by (c) a tree and (d) distracting street texture can be handled.

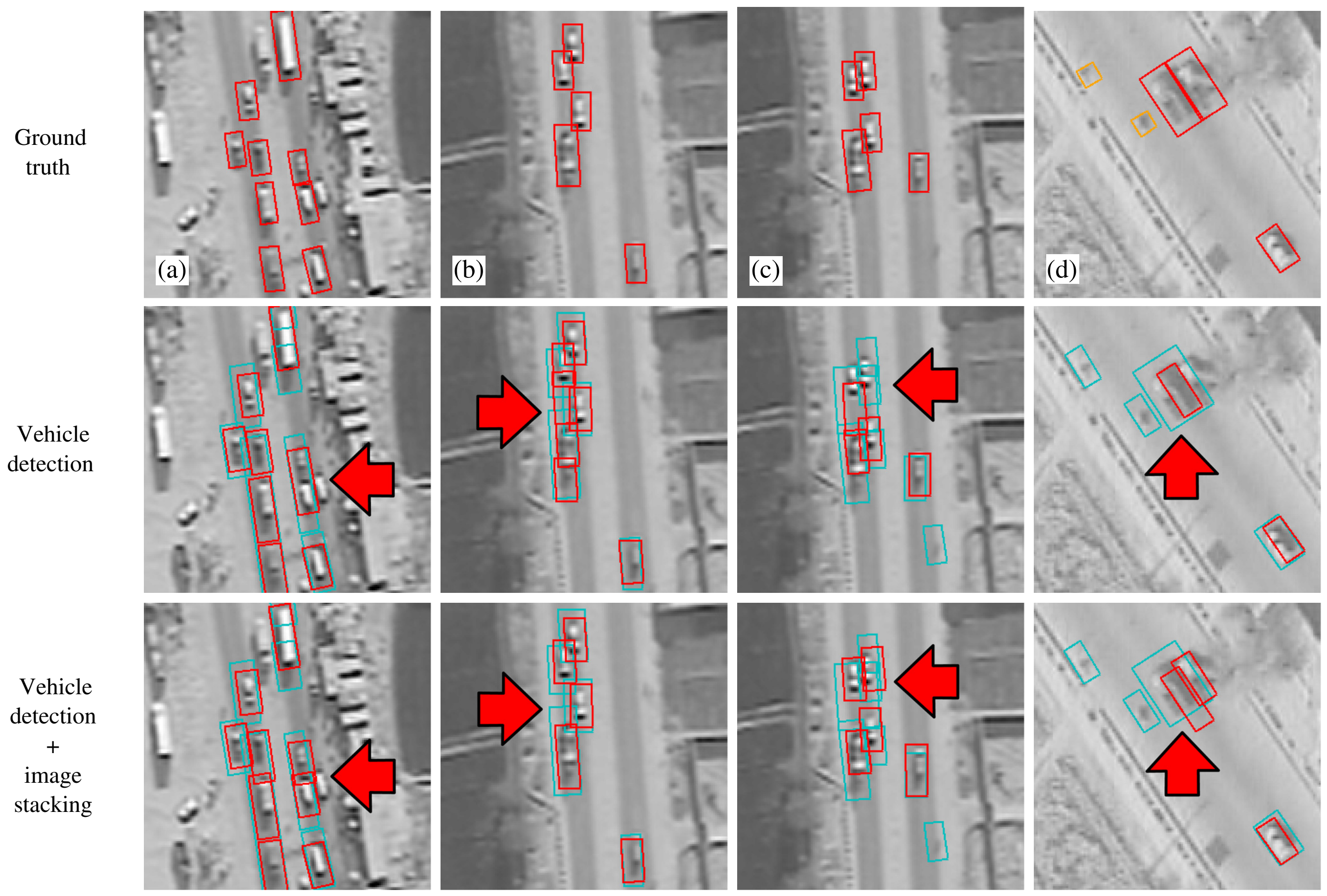


Fig. 14 Examples taken from the qualitative evaluation of the proposed image stacking approach applied to moving vehicle detection. Merged, FP, or incorrectly located detections can be handled, if there is a relative velocity difference between the vehicles (a)–(c). Furthermore, FN detections due to occlusion by (d) a tree can be avoided.

(Sec. 4.4) claims between 28 ms, if the accumulation image is used, and up to 850 ms, if the stack is arranged as a circular buffer. Optimization can be achieved by fast, approximated, or incremental calculation of the median pixel values in the circular buffer.[63,64] Furthermore, we do not perform any parallel processing of the image stacks, which is expected to strongly reduce the runtime.

6 Conclusions and Outlook

The main goal of our proposed image stacking approach is to remove disturbing image structures from the background of moving objects on the ground observed by airborne cameras in top view. Those structures are usually not moving and can thus be smoothed to isolate the observed object from the background. Especially, model-free object segmentation can benefit from this approach since object contours can be well-isolated by image stacking. There is less benefit for sliding window-based vehicle detection since the assumptions that are made implicitly (e.g., about object size or object width/length ratio) tackle similar problems as the image stacking approach. Furthermore, imprecise image registration of moving image regions that are used for image stacking is the most prominent limiting factor for vehicle detection as vehicles can be blurred affecting the classifier's confidence. However, improvement may be possible when using deep learning-based local feature matching for image registration with higher accuracy.[65]

There are several potential applications for image stacking besides improved object detection and segmentation. Among these applications is super-resolution for moving objects, generating appearance templates without background for visual object tracking or reidentification, or temporal filtering to suppress image noise, compression artifacts, or artifacts of a disturbed wireless connection.

Acknowledgments

The research reported in this contribution was funded by the Technical Center for Information Technology and Electronics (WTD 81) of the German Federal Office for Armament and Procurement.

References

1. B. Zitová and J. Flusser, "Image registration methods: a survey," *Image Vision Comput.* **21**, 977–1000 (2003).
2. R. P. Kleihorst, *Noise Filtering of Image Sequences*, Dissertation, Delft University of Technology, The Netherlands (1994).
3. T. Müller and M. Müller, "CART IV: improving camouflage assessment with assistance methods," *Proc. SPIE* **7662**, 76620R (2010).
4. W. Li, J. Zhang, and Q. Dai, "Exploring aligned complementary image pair for blind motion deblurring," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'11)*, pp. 636–647 (2011).
5. S. Cho, H. Cho, and Y. Tai, "Registration based non-uniform motion deblurring," *Comput. Graphics Forum* **31**, 2183–2192 (2012).

6. X. Zhu, F. Šroubek, and P. Milanfar, "Deconvolving PSFs for a better motion deblurring using multiple images," *Lect. Notes Comput. Sci.* **7576**, 636–647 (2012).
7. J. Migdal, T. Izo, and C. Stauffer, "Moving object segmentation using super-resolution background models," in *Proc. of the IEEE Int. Conf. on Computer Vision Workshops (ICCVW'05)* (2005).
8. V. Reilly, H. Idrees, and M. Shah, "Detection and tracking of large number of targets in wide area surveillance," *Lect. Notes Comput. Sci.* **6313**, 186–199 (2010).
9. S. Farsiu et al., "Fast and robust multiframe super resolution," *IEEE Trans. Image Process.* **13**, 1327–1344 (2004).
10. C. Liu and D. Sun, "A Bayesian approach to adaptive video super resolution," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'11)*, pp. 209–216 (2011).
11. H. Zhang and L. Carin, "Multi-shot imaging: joint alignment, deblurring, and resolution-enhancement," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'14)*, pp. 2925–2932 (2014).
12. Z. Ma et al., "Handling motion blur in multi-frame super-resolution," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'15)*, pp. 5224–5232 (2015).
13. A. Letienne et al., "Fast super-resolution on moving objects in video sequences," in *Proc. of the European Signal Processing Conf. (EUSIPCO'08)* (2008).
14. A. van Eekeren, K. Schutte, and L. van Vliet, "Multiframe super-resolution reconstruction of small moving objects," *IEEE Trans. Image Process.* **19**, 2901–2912 (2010).
15. M. Teutsch and W. Krüger, "Spatio-temporal fusion of object segmentation approaches for moving distant targets," in *Proc. of the Int. Conf. on Information Fusion (FUSION)* (2012).
16. M. Teutsch, "Moving object detection and segmentation for remote aerial video surveillance," Dissertation, Karlsruhe Institute of Technology, Germany (2015).
17. A. Perera et al., "Multi-object tracking through simultaneous long occlusions and split-merge conditions," in *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR'06)* (2006).
18. J. Xiao et al., "Vehicle detection and tracking in wide field-of-view aerial video," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'10)* (2010).
19. I. Saleemi and M. Shah, "Multiframe many-many point correspondence for vehicle tracking in high density wide area aerial videos," *Int. J. Comput. Vision* **104**, 198–219 (2013).
20. M. Siam and M. ElHelw, "Robust autonomous visual detection and tracking of moving targets in UAV imagery," in *Proc. of the IEEE Int. Conf. on Signal Processing (ICSP'12)* (2012).
21. M. Teutsch and W. Krüger, "Detection, segmentation, and tracking of moving objects in UAV videos," in *Proc. of the IEEE Int. Conf. on Advanced Video and Signal-Based Surveillance (AVSS'12)* (2012).
22. M. Siam and M. ElHelw, "Enhanced target tracking in UAV imagery with P-N learning and structural constraints," in *Proc. of the IEEE Int. Conf. on Computer Vision Workshops (ICCVW'13)* (2013).
23. P. Kent et al., "Robust background subtraction for automated detection and tracking of targets in wide area motion imagery," *Proc. SPIE* **8546**, 85460Q (2012).
24. T. Pollard and M. Antone, "Detecting and tracking all moving objects inwide-area aerial video," in *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW'12)* (2012).
25. Z. Zheng et al., "A novel vehicle detection method with high resolution highway aerial image," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **6**(6), 2338–2343 (2013).
26. H.-Y. Cheng, C.-C. Weng, and Y.-Y. Chen, "Vehicle detection in aerial surveillance using dynamic Bayesian networks," *IEEE Trans. Image Process.* **21**, 2152–2159 (2012).
27. S. Sahli et al., "Robust vehicle detection in aerial images based on salient region selection and superpixel classification," *Proc. SPIE* **8020**, 80200L (2011).
28. H. Meuel et al., "Superpixel-based segmentation of moving objects for low bitrate ROI coding systems," in *Proc. of the IEEE Int. Conf. on Advanced Video and Signal-Based Surveillance (AVSS)* (2013).
29. X. Cao et al., "Vehicle detection and motion analysis in low-altitude airborne video under urban environment," *IEEE Trans. Circuits Syst. Video Technol.* **21**(10), 1522–1533 (2011).
30. A. Gaszczak, T. Breckon, and J. Han, "Real-time people and vehicle detection from UAV imagery," *Proc. SPIE* **7878**, 78780B (2011).
31. R. Pelapur et al., "Persistent target tracking using likelihood fusion in wide-area and full motion video sequences," in *Proc. of the Int. Conf. on Information Fusion (FUSION)*, pp. 2420–2427 (2012).
32. R. Lin et al., "Airborne moving vehicle detection for video surveillance of urban traffic," in *Proc. of the IEEE Intelligent Vehicles Symp. (IV)* (2009).
33. P. Liang et al., "Multiple kernel learning for vehicle detection in wide area motion imagery," in *Proc. of the Int. Conf. on Information Fusion (FUSION)* (2012).
34. M. Teutsch and W. Krüger, "Robust and fast detection of moving vehicles in aerial videos using sliding windows," in *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW'15)* (2015).
35. A. Ibrahim et al., "Moving objects detection and tracking framework for UAV-based surveillance," in *Proc. of the Fourth Pacific-Rim Symp. on Image and Video Technology* (2010).
36. J. Gleason et al., "Vehicle detection from aerial imagery," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)* (2011).
37. Q. Li et al., "Real-time highway traffic information extraction based on airborne video," in *Proc. of the Int. IEEE Conf. on Intelligent Transportation Systems (ITSC'09)* (2009).
38. P. Luo et al., "Stationary vehicle detection in aerial surveillance with a UAV," in *Proc. of the Int. Conf. on Information Science and Digital Content Technology (ICIDT'12)* (2012).
39. X. Shi et al., "Context-driven moving vehicle detection in wide area motion imagery," in *Proc. of the Int. Conf. on Pattern Recognition (ICPR'12)* (2012).
40. J. Prokaj and G. Medioni, "Persistent tracking for wide area aerial surveillance," in *Proc. of the 2014 IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR'14)* (2014).
41. S. Ali, V. Reilly, and M. Shah, "Motion and appearance contexts for tracking and re-acquiring targets in aerial videos," in *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR'07)* (2007).
42. O. Mise and T. Breckon, "Super-resolution imaging applied to moving targets in high dynamics scenes," *Proc. SPIE* **8899**, 889917 (2013).
43. J. Shi and C. Tomasi, "Good features to track," in *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR'94)* (1994).
44. W. Förstner and E. Gülch, "A fast operator for detection and precise location of distinct point, corners and centres of circular features," in *Proc. of the ISPRS Conf. on Fast Processing of Photogrammetric Data*, pp. 281–305 (1987).
45. C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. of the Fourth Alvey Vision Conf.* (1988).
46. R. Hartley and A. Zisserman, *Multiple-View Geometry in Computer Vision*, 2nd ed., Cambridge University Press, Cambridge, United Kingdom (2004).
47. M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM* **24**, 381–395 (1981).
48. B. S. Everitt et al., *Cluster Analysis*, 5th ed., Wiley, Hoboken, New Jersey (2011).
49. M. Teutsch, W. Krüger, and J. Beyerer, "Evaluation of object segmentation to improve moving vehicle detection in aerial videos," in *Proc. of the IEEE Int. Conf. on Advanced Video and Signal-Based Surveillance (AVSS)* (2014).
50. P. Dollár et al., "Integral channel features," in *Proc. of the British Machine Vision Conf. (BMVC'09)* (2009).
51. F. Y. Shih, *Image Processing and Pattern Recognition: Fundamentals and Techniques*, Wiley, Hoboken, New Jersey (2010).
52. P. Arbelaez et al., "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.* **33**, 898–916 (2011).
53. R. Achanta et al., "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**, 2274–2282 (2012).
54. H. Shen et al., "Moving object detection in aerial video based on spatiotemporal saliency," *Chin. J. Aeronaut.* **26**, 1211–1217 (2013).
55. J. A. Hartigan, *Clustering Algorithms*, Wiley, Hoboken, New Jersey (1975).
56. R. T. Collins, X. Zhou, and S. K. Teh, "An open source tracking testbed and evaluation web site," in *Proc. of the IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance (PETS)* (2005).
57. U.S. AFRL, "Columbus large image format (CLIF) dataset," 2007, https://www.sdms.afrl.af.mil/index.php?collection=clif2007 (23 July 2017).
58. U.S. AFRL, "Wright-Patterson air force base (WPAFB) dataset," (2009), https://www.sdms.afrl.af.mil/index.php?collection=wpafb2009 (23 July 2017).
59. G. Csurka, D. Larlus, and F. Perronnin, "What is a good evaluation measure for semantic segmentation?" in *Proc. of the British Machine Vision Conf. (BMVC)* (2013).
60. M. Everingham et al., "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vision* **88**, 303–338 (2010).
61. N. Japkowicz and M. Shah, *Evaluating Learning Algorithms: A Classification Perspective*, Cambridge University Press, Cambridge, United Kingdom (2011).
62. R. Kasturi et al., "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: data, metrics, and protocol," *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(2), 319–336 (2009).
63. R. J. Tibshirani, "Fast computation of the median by successive binning," arXiv:0806.3301v1 [stat.CO] (2008).
64. J. Cadenas et al., "Fast median calculation method," *Electron. Lett.* **48**, 558–560 (2012).
65. H. Altwaijry et al., "Learning to match aerial images with deep attentive architectures," in *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR'16)* (2016).

Michael Teutsch received his diploma degree in computer science and PhD from the Karlsruhe Institute of Technology (KIT) in 2009 and 2014, respectively. From 2009 to 2016, he worked as a research scientist at the Fraunhofer IOSB, Karlsruhe, Germany. Since 2016, he has been with Hensoldt Optronics, Oberkochen, Germany. His research interests include computer vision, visual surveillance, object detection, object tracking, and machine learning. He has authored or coauthored more than 20 scientific publications.

Wolfgang Krüger received his diploma degree in computer science and his PhD from the University of Karlsruhe, Germany, in 1986 and 1991, respectively. Since 1986, he has been a research scientist at the Fraunhofer IOSB, Karlsruhe. His research interests include computer vision, analysis of unmanned aerial vehicle video data, image registration, motion detection, object detection, visual tracking, and machine learning.

Jürgen Beyerer is a full professor for informatics at the Institute for Anthropomatics and Robotics at the KIT since March 2004 and director of the Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB in Ettlingen, Karlsruhe, Ilmenau, and Lemgo. His research interests include automated visual inspection, signal and image processing, variable image acquisition and processing, active vision, metrology, information theory, fusion of data and information from heterogeneous sources, system theory, autonomous systems, and automation.