Detection and classification of moving objects from UAVs with optical sensors

Michael Teutsch, Wolfgang Krüger, Norbert Heinze
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB)
Fraunhoferstr. 1, 76131 Karlsruhe, Germany

ABSTRACT
Small and medium sized UAVs like German LUNA have long endurance and define in combination with sophisticated image exploitation algorithms a very cost efficient platform for surveillance. At Fraunhofer IOSB, we have developed the video exploitation system ABUL with the target to meet the demands of small and medium sized UAVs. Several image exploitation algorithms such as multi-resolution, super-resolution, image stabilization, geocoded mosaiking and stereo-images/3D-models have been implemented and are used with several UAV-systems. Among these algorithms is the moving target detection with compensation of sensor motion. Moving objects are of major interest during surveillance missions, but due to movement of the sensor on the UAV and small object size in the images, it is a challenging task to develop reliable detection algorithms under the constraint of real-time demands on limited hardware resources. Based on compensation of sensor motion by fast and robust estimation of geometric transformations between images, independent motion is detected relatively to the static background. From independent motion cues, regions of interest (bounding-boxes) are generated and used as initial object hypotheses. A novel classification module is introduced to perform an appearance-based analysis of the hypotheses. Various texture features are extracted and evaluated automatically for achieving a good feature selection to successfully classify vehicles and people.

Keywords: Video exploitation, UAV, moving object detection, tracking, classification, surveillance

1. INTRODUCTION
An important objective of surveillance is the detection of activity. This can be moving vehicles or persons. By using tactical UAVs a broad area can be monitored for several hours by very few persons without endangering their lives in patrolling. But for surveying large areas, wide angle optics has to be used. So, object-related activity is hardly expressed in pixel-size. Keeping in mind that with small UAVs we have an endurance of several hours, the photo interpreter has a challenging task with fatigue as major problem.

We think that supporting the photo interpreter with appropriate automatic image exploitation algorithms can improve the detection rate under the condition of missions lasting several hours. Hence, we have developed algorithms to detect small moving objects in UAV video imagery. To have large areas in focus, comparable wide optics is used, so that object-size and movement become small. Most challenging is the detection of sub-pixel motion of small, low contrast objects. Additionally, detection of objects often has to be done in urban areas under tactical UAV conditions. This means low flight height and stereo effects due to the 3D structure of urban buildings. More challenges are interlaced video of existing UAVs and effects of the data-transmission/compression. Our detection algorithm is described in section 3.

Furthermore, it can be important to know if the detected movements are originating from persons or vehicles. On basis of the moving target detection we have developed a classification layer to separate the two classes people consisting of single or multiple persons, bicyclists, and motorcyclists, as well as vehicles consisting of cars, trucks, and busses. Our approach for classification and the results are described in the section 4. Related work for both the moving target detection and classification will be presented directly in section 3 and 4. All developed and tested algorithms are integrated into our image exploitation-system for UAVs ABUL, so that we have the possibility to evaluate the benefits and effects with photo-interpreters and can get feedback for further improvements. The moving object detection is already integrated in ABUL, while the classification integration is under development. The system ABUL and its application is described in more detail in section 2.

Further author information: Email: {michael.teutsch, wolfgang.krueger, norbert.heinze}@iosb.fraunhofer.de
2. IMAGE EXPLOITATION SYSTEM ABUL

We have developed a video exploitation system ABUL as seen in Fig. 1, mainly meeting the demands of small UAVs like German LUNA. LUNA has several cameras of different focal length mounted together on the UAV. For this research work, LUNA flew at an altitude of about 400 m using one wide angle camera directed nearly perpendicular to the ground. With the capability to read STANAG 4609 video-files and to receive STANAG 4609 video-streaming data, also video from other interoperable UAV-systems like Heron, Predator, etc. can be processed. Specialized interfaces are integrated for some UAV systems and sensors including the MiSAR system, which is the SAR system for tactical UAVs weighting only 4 kg.\(^1\)\(^,2\)

Several image exploitation algorithms were developed and integrated and can easily be used by photo interpreters. The following algorithms are currently integrated into the system:

1. **Multi-resolution:** Images of different focal length are processed to generate data fusion results. With the LUNA sensor generating images of different focal length, we compute high-resolution images of approximately 2000 × 2000 pixels (see Fig. 2) of the whole flight-path from the video-imagery transmitted in real-time. This algorithm is based on the video-image only and doesn’t need the INS data, so it is not depended on an expensive high-precision INS.

2. **Super-resolution:** Due to compression or effects of the transmission, images have often a reduced resolution. Super-resolution computes from image-sequences images of better quality and resolution.

3. **Super-resolution with emphasis on SAR sequences:** SAR image sequences are processed to generate high resolution products with less noise.

4. **Image stabilization:** Processing the video stream, the displayed output is stabilized in real-time, enabling the operator to achieve optimal results.

5. **Real-time detection of moving objects with moving sensor:** On moving platforms moving targets are automatically detected, though the sensor platform is in motion itself. The underlying algorithm performs with a very low false alarm rate (FAR) despite its high sensitivity.

6. **Geo-coded mosaicking:** Processing the video stream, the video images are stitched to panorama images in real-time, which provide excellent overview.

7. **Stereo images / 3D-models:** The video stream is automatically processed in order to provide three-dimensional views.
This gives us the chance to evaluate the system in close collaboration with relevant users. So, discussion and feedback with users can take place resulting in new ideas for algorithms and tools.\textsuperscript{3} Test systems are supplied to several departments of the German Bundeswehr and systems are integrated in UAV-ground-control-stations in Afghanistan as seen in Fig. 3, so that the performance of these experimental systems can be gradually improved by incorporating feedbacks and actual scientific results.
3. MOVING TARGET DETECTION

Moving object detection in this paper and previous work\(^2\) is based on robust tracking of local image features and image registration. Moving objects are detected at locations, where local image motion does not coincide with the global image transformation used for registration. That approach is for example effective in video-surveillance scenarios in which a moving camera observes a distant scene. Since the differences in scene in depth are small relative to the distance of the observing camera, the scene may be approximated by a ground plane, and a homography\(^4\) can be used as the global transformation to register different image frames.

Similar ground plane assumptions have been used for a fairly long time for vision-based driver assistance to detect obstacles from the motion compensated difference of successive image frames.\(^5,6\) The idea of motion compensated difference images (subtraction of successive images registered to a common ground plane) has also been applied for aerial image exploitation to detect changes and moving objects.\(^7,8\) Plane+parallax approaches which check constraints on point correspondences between two or more images\(^9-11\) have been proposed to handle low altitude imagery with prominent 3D structure.

Object detection in this paper employs the ground plane assumption because our image data is mainly from UAVs operating at higher altitudes. Instead of motion compensated difference images we use relative image velocities estimated from point feature tracking\(^12\) and image registration.\(^13\) The reason is that we have been able to achieve much more reliable and sensitive detections compared to our work using motion compensated difference images.\(^8\) Our approach does not require camera calibration and is able to run in real-time on standard PC-hardware. Since it relies on relative motion information it is largely independent of object appearance. Experiments with a huge amount of real image data from UAVs have shown that our implementation yields very rare false detections while still being sensitive to small object motion. Moreover, it is able to cope with the compression and interlacing artifacts contained in the transmitted UAV image streams.

Our algorithm to detect moving objects consists of the following steps:

1. Computation of point correspondences between successive image frames and estimation of the global image transformation. Since estimation has to be robust with respect to outliers, random sampling (RANSAC) with subsequent refinement is used. Fig. 4a shows correspondences from current image to a previous image (3 frames apart).

2. Estimation of relative image velocities from point feature tracks and the global transformation calculated in step 1. This step compensates the dominating image motion caused by the moving platform (Fig. 4a and 4b).

3. Detection of significant relative motion. After compensation of global image motion, thresholding is used to separate velocities estimated at moving objects from those at ground plane points (cyan vectors vs. magenta vectors in Fig. 4b).

4. Computation of bounding-boxes for classification layer. Since the aim of our application is to detect small moving objects like distant vehicles or people, we decided to use two types of bounding-boxes: Bounding-boxes for single detections (Fig. 5a), and bounding-boxes for pairs of compatible velocity vectors (Fig. 5b). The motivation for pairing compatible velocity vectors is to find front and back of the same vehicle. In crowded scenes this may also lead to wrong hypotheses for bounding-boxes of vehicles. Since relative motion information alone is not sufficient to resolve such cases, additional appearance cues will be employed by the classification layer.

The next section describes how appearance-based classification processes the bounding-boxes computed by this motion detection algorithm.
Figure 4. Computation of point correspondence and compensation of UAV-motion to find independent motion.
Figure 5. Computation of bounding-boxes for classification layer. Single detections (magenta) and detection pairs (cyan).
4. MOVING TARGET CLASSIFICATION

In most related work about either people or vehicle detection and classification from UAVs, motion is not a
key feature to find objects. This is good on the one hand, because in some applications static objects are of
interest, too. On the other hand, it is difficult to find reliable approaches especially when working with images
of high ground sampling distances (GSD) where objects cover only few pixels. Lavigne et al.\textsuperscript{14} detect vehicles in
images with a GSD of 0.118 $\text{m/pixel}$ by extracting SIFT-features, classifying detections with a SVM and clustering
the results with an unsupervised affinity propagation clustering algorithm. A combination of Haar-features,
histograms of oriented gradients (HOGs) and Local Binary Pattern (LBP) features is used by Nguyen et al.\textsuperscript{15} to
perform on-line boosting classification and clustering with mean shift. This way, vehicles can be found in images
with a GSD of 0.08 respectively 0.1 $\text{m/pixel}$. Vehicles and persons are detected and separated by Gaszczak et al.\textsuperscript{16}
using cascaded Haar classifiers in optical and infrared image sequences. The images are taken from a UAV at an
altitude of 400 $\text{m}$ and 45 degrees of camera view angle. Reilly et al.\textsuperscript{17} detect persons with a human blob detector
using optional geometric constraints coming from UAV meta information such as altitude, position, orientation
as well as focal length and time. A combination of wavelet features processed by a SVM leads to a separation
of clutter and persons. Shadow biometrics is used by Iwashita et al.\textsuperscript{18} to find people in UAV data. But up to
now, a fixed camera with background subtraction and low GSD of 0.01 – 0.02 $\text{m/pixel}$ is necessary. Finally, moving
objects are detected and tracked by Xiao et al.\textsuperscript{19} HOGs are extracted from the detections to separate vehicles
and persons with a SVM. The correct classification rates are between 65 and 95\%.

With the introduction of classification, we want to achieve two aims: distinguish between different object
classes and support the fusion process of bounding-boxes, which are spatially close to each other. Since the
moving target detection is very reliable and produces nearly no clutter detections, it is not necessary to consider
clutter as a desired class for the classification process. Thus, we only concentrate on two classes:

- \textit{Moving people}: Single or multiple persons, bicyclists, motorcyclists.
- \textit{Moving vehicles}: Cars, trucks, buses.

Distinguishing between these two classes is still a sophisticated problem, as typical object appearance covers
only a few pixels in size. For our used UAV LUNA flying at an altitude of about 400 $\text{m}$ and a GSD of 0.3 $\text{m/pixel}$
in horizontal and vertical direction, a standard car with length 4.2 $\text{m}$ and width 1.8 $\text{m}$ covers 14 $\times$ 6 pixels.
Semantically, bike- and motorcyclists could be interpreted as vehicles, too. But their appearance is much more
similar to the one of people. The classification module consists of three steps: pre-processing, feature extraction
and classification, and post-processing. In the follow-up, these steps will be described in detail.

4.1 Pre-processing

For the subsequent feature extraction and classification step, we aim to get normalized detection hypotheses with
respect to rotation and scale. Therefore, we use the motion direction, which is directly delivered by the moving
target detection, and the GSD, which is calculated by using meta-data coming from the UAV itself.

Motion direction is given as two-dimensional motion vector $\vec{v} = (v_x, v_y)$. To get the desired rotation angle
$\alpha$ between $\vec{v}$ and reference vector $\vec{r} = (0, 1)$, we use the arctangent-function $\text{atan2}$. Rotation origin is center
$c = (x_c, y_c)$ of the bounding-box as we expect the object to be right in the middle of the bounding-box. Like in
standard image processing rotation approaches, we use inverted $\alpha$ to get the source pixel position $src$ of a wanted
destination pixel position $dst$. The source will be a sub-pixel position and its intensity value can be calculated
using bilinear interpolation. This leads to the following equations for each destination pixel position $dst$ in the
rotated bounding-box image:

\begin{align}
\alpha &= \text{atan2}(1, 0) - \text{atan2}(v_y, v_x) \\
x_{src} &= \cos(-\alpha) \cdot (x_{dst} - x_c) - \sin(-\alpha) \cdot (y_{dst} - y_c) + x_c \\
y_{src} &= \sin(-\alpha) \cdot (x_{dst} - x_c) + \cos(-\alpha) \cdot (y_{dst} - y_c) + y_c
\end{align}
If the UAV altitude varies, also the object scaling differs. We take this into account by calculating the horizontal and vertical ground sampling distances $GSD_x$ and $GSD_y$ using a basic elevation model. The ratio between current GSD and a reference ground sampling distance $GSD_{ref}$ will be the scaling factor. This leads to the following modification of equation 2 and 3:

$$x_{src} = \frac{GSD_{ref}}{GSD_x} \cdot \cos(-\alpha) \cdot (x_{dst} - x_c) - \frac{GSD_{ref}}{GSD_y} \cdot \sin(-\alpha) \cdot (y_{dst} - y_c) + x_c \quad (4)$$

$$y_{src} = \frac{GSD_{ref}}{GSD_x} \cdot \sin(-\alpha) \cdot (x_{dst} - x_c) + \frac{GSD_{ref}}{GSD_y} \cdot \cos(-\alpha) \cdot (y_{dst} - y_c) + y_c \quad (5)$$

Thus, we achieve a normalized object appearance with respect to rotation and scale as seen in Fig. 6. This is necessary for the feature extraction since not all of the considered features are invariant towards rotation and scale. The benefit of scale normalization for classification will be pointed out later in Table 1 in section 4.4.

### 4.2 Feature extraction and classification

Feature extraction and classification is done with a framework which has already been used successfully for ship classification in TerraSAR-X images\textsuperscript{20} and classification of small boats in infrared image sequences.\textsuperscript{21} In this work, we skipped the introduction of precise object segmentation, which obviously leads to worse classification results. In SAR and IR images, object appearances are generally represented by high pixel intensity values. Usually, this is not the case in visual-optical gray-scale images, which we use here. So, it wasn’t possible to adapt the segmentation approaches applied in the framework up to now. This is a potential drawback and will be discussed later in the outlook in section 5.

Fig. 7 shows the framework in context of the whole moving target detection and classification system presented here. It particularly consists of the modules feature extraction, descriptor setup, and classification. Feature extraction is characterized by the calculation of a big database of 386 different features. These features are extracted from given pre-processed detection hypotheses and subdivided in eight feature classes:

- **Invariant moments** such as Hu moments.
- Texture features calculated on co-occurrence matrices.
- General texture features such as gray-value mean or variance.
- Features calculated on the segmented object blob such as central moments.
Textural relations between the *kernel* (segmented object blob) and its surrounding area.

- Textural features like mean, variance, etc. calculated and compared row-wise to keep spatial information.
- *Gradient*-based features.
- Features based on the calculation of *Local Binary Pattern* (LBP).

Please refer to our recent publication\(^\text{21}\) for a more detailed description of the different features. The processing time to calculate all of them in each time step is by far too high for a real-time application like ABUL. Thus, we calculate them only during the classifier training stage for a feature evaluation and selection.

With manually labeled training data, an automatic and data-driven approach for feature evaluation and selection is performed. Each feature is analyzed individually for its separability for the given classification task. A greedy algorithm is used to determine the feature combination with highest separability. Potential interdependence between features is considered by setting up a covariance matrix and taking it into account for the greedy algorithm. In this specific classification task of separating vehicles and people in UAV gray-scale video data, the automatic feature selection discovered a combination of 7 features originating from texture analysis, blob analysis, and row analysis to be the best for processing the training data. Hence, only very general features with little spatial information have been chosen due to high object distance and weak shape information, which made the system rejecting for example gradient-based features. Finally, we not only identified an appropriate feature vector (*descriptor*), but also many features with weak separability. From now on their calculation can be skipped to save processing time.

For classification, a database of 4067 training samples has been used to train a *Support Vector Machine* (SVM). Each training sample is given by the 7-dimensional descriptor including the manually set label (vehicle or person). In subsequent classifier evaluation, we found out that *k*-Nearest-Neighbor (*k*-NN) classifier outperforms the SVM for this classification task. For a given feature vector, the *k*-NN classifier searches for the closest *k* neighboring descriptors of the training data. If more than \( \frac{k}{2} \) of them are labeled as vehicle, the given descriptor will also be classified as vehicle. In the final application we use a 9-NN classifier. Evaluation results and correct classification rates will be presented in section 4.4.

### 4.3 Post-processing

As already mentioned in the beginning of this section, the second aim of classification besides separation of people and vehicles is to support bounding-box fusion. The moving target detection algorithm tends to detect...
moving target detection classification bounding-boxes for fusion check
rotation by angle of motion direction of first bounding-box
find overlap overlap > t\textsubscript{O} fusion inverse rotation

Figure 8. Example for fusion of two bounding-boxes based on the three fusion criteria.

object’s front and back with two different bounding-boxes. Performing a fusion based on comparison of motion directions right after detection is possible but can lead to an undesirable fusion of people and vehicles close to each other. In case of a vehicle overtaking a pedestrian, the motion direction is barely the same, so a false fusion will be the consequence. Thus, the moving target detection algorithm created fusion hypotheses of bounding-boxes spatially close to each other as seen in Fig. 5. These hypotheses will be analyzed now with further knowledge after classification to improve the fusion process. Therefore, we use three criteria for the fusion of two bounding-boxes:

1. Motion directions difference must be smaller than threshold \( t_M \).
2. Object classes must be the same.
3. Overlapping area of the bounding-boxes must be bigger than threshold \( t_O \).

Since people motion direction can change quickly and more spontaneously than vehicle motion direction, we chose a bigger \( t_M \) for people than for vehicles. First and second criterion can be analyzed straight forward, but for the third criterion, we perform another rotation of bounding-boxes because overlapping calculation with non-paraxial bounding-boxes is quite difficult. The whole process is visualized in Fig. 8. After moving target detection and classification with vehicles in red and people in yellow bounding-boxes, we randomly choose two bounding-boxes, which fulfill the first and second criterion. They are rotated by the motion direction angle of the first bounding-box using equations 2 and 3. After aligning the rotated bounding-boxes to the axes of the rotated image coordinate system, we can easily determine the overlapping area (slight red) and if it is above threshold \( t_O \), the bounding-boxes are fused. Finally, the fused bounding-box is rotated back to the original image coordinate system.

4.4 Experimental results
In our experiments, we used two different image sequences coming from the same UAV sensor with an image resolution of \( 720 \times 576 \) pixels. The first scene has a GSD of \( 0.345 \frac{m}{pixel} \) and consists of 1201 single images with 33270 moving target detections in summary. We manually labeled 4067 detections distributed across the whole sequence either for being people or vehicle. This scene was used for classifier training exclusively. The second image sequence has 733 images with 4461 detections and a GSD of about \( 0.225 \frac{m}{pixel} \). This scene was manually
labeled, too, and taken for evaluation only. Among the 4461 detections, there are 3215 vehicles and 1246 people. The two example sequences differ in UAV altitude, object shadow, and number of objects. Furthermore, it is most likely, that no object is appearing in both image sequences.

With the second scene, we tested several different classifiers and evaluated the effect of considering GSD or not. The results can be seen in Table 1. Both $k$-NNs outperformed the SVM. This might have been the case because choosing training data for the SVM appeared to be pretty difficult in this application and maybe it wasn’t chosen properly. Another hint for this assumption is the noticeable benefit of GSD consideration for $k$-NN but not for SVM. We also tested $k$-NNs with $k > 9$, but this had no noteworthy influence to the results. Among the features chosen by the automatic feature selection algorithm (see section 4.2) there are general texture features such as mean gray-value and variance in the bounding-box area. This is based on the assumption that vehicles produce higher gray-value variance than people, which is valid in most cases. Fig. 9 shows four example images, while Fig. 9a and 9b are taken from the first scene and Fig. 9c and 9d from the second. Bounding-boxes with vehicles are displayed in red and people in yellow. All not detected objects are moving only very slowly or not at all. In Fig. 9d we see a misclassified person. This occurred due to violation of the assumption that vehicles produce higher gray-value variance than people. The buildings close to the person cause this violation. Additionally, we see that false fusion (see Fig. 9a and 9b) and the detection of shadows as part of a moving object (see Fig. 9c) are further starting-points to improve the current approach. On a standard PC without any process parallelization, the runtime for moving target classification is between 8 ms for few objects and 40 ms for approximately 10 or more objects.

### 5. CONCLUSIONS AND FUTURE WORK

We presented a processing chain for moving target detection and classification in UAV image sequences with a GSD of about 0.3 m/pix. Independent motion with sub-pixel accuracy is detected, tracked and clustered to generate bounding-boxes for a subsequent classification module. Since the moving target detection works reliably and stable, classification doesn’t need to separate between object and clutter but only between vehicles and people. The classification begins with pre-processing to get a normalized object appearance with respect to rotation and scale. On the normalized bounding-boxes, an object descriptor is created using an automatic feature selection algorithm. With a 9-NN classifier we achieved the best correct classification rate of 96.08%. In post-processing, we perform a fusion of bounding-boxes with same class, similar motion direction, and sufficient overlapping. The moving target detection and classification approach is very fast and doesn’t need any camera parameters or meta information.

In future work, we want to introduce a precise object segmentation to extract object contours. This will improve all steps of the classification module reducing potential false bounding-box fusion as well as diminishing the undesired background area for feature extraction. Additionally, further discrimination between passenger cars and larger vehicles is under consideration.

### ACKNOWLEDGMENTS

We would like to thank WTD 81 (Greding), BWB-L1.1 (Koblenz), KSA/SGA (Gelsdorf), EADS Germany (Ulma/Friedrichshafen), and EMT (Penzberg) for supporting our work.
Figure 9. Some examples for moving target detection and classification including the fusion process. Vehicles are displayed in red and people in yellow. Images (a) and (b) are coming from the first scene with a GSD of $0.345 \text{ m/pixel}$, images (c) and (d) from the second scene with a GSD of $0.225 \text{ m/pixel}$. Not detected objects are moving only very slowly or not at all.
REFERENCES


