

# Integration of UAVs in Urban Search and Rescue Missions\*

Hartmut Surmann<sup>1</sup>, Rainer Worst<sup>2</sup>, Tim Buschmann<sup>1</sup>, Artur Leinweber<sup>1</sup>,  
Alexander Schmitz<sup>1</sup>, Gerhard Senkowski<sup>1</sup>, and Niklas Goddemeier<sup>3</sup>

**Abstract**—This technical report is about the architecture and integration of commercial UAVs in Search and Rescue missions. We describe a framework that consists of heterogeneous UAVs, a UAV task planner, a bridge to the UAVs, an intelligent image hub, and a 3D point cloud generator. A first version of the framework was developed and tested in several training missions in the EU project TRADR.

## I. INTRODUCTION

Portable unmanned aircraft vehicles (UAVs  $< 2$  kg) are widely used by military, administrative, and commercial organizations, but are also used by consumers (hobbyists). In Europe and the United States, UAVs are regulated as remotely piloted aircrafts; they are sometimes called unmanned aviation vehicles or in lay terms drones. They can be used for a variety of applications such as reconnaissance, surveillance, and search. Large UAVs ( $>5$  kg) are applied for logistics, e.g., delivery of light supplies and equipment, for research, for data collection, e.g., agriculture, land survey, and weather monitoring, for communication, and for aerial photography.

In Europe, consumer and commercial UAVs are regulated by the European Aviation Safety Agency (EASA) [1]. European rules will come into effect in 2019/20. Until then, e.g., in Germany the Drohnen-Verordnung from 2017 is valid. Above a weight of 250 g, UAVs need a liability insurance, a sign plate, and have to fly below 100 meters. An operator flight license is necessary for UAVs above 2 kg and at some places it is not allowed to fly at all, e.g., hospitals, military facilities. Even small UAVs ( $<1$  kg) are restricted to a maximum of 1 km altitude and 2 km range from the base operator.

Although large UAVs ( $< 25$  kg) have been used successfully in natural and urban disasters, small portable UAVs are recently being used by search and rescue (SAR) experts [2], [3], [4]. Nevertheless, the full potential of small drones can only be developed by defining a suitable workflow and integrating them into the procedures of rescue workers. At this point starts the paper. After this introduction, the next chapter II gives a short state of the art. Chapter III presents our suggestion for an UAV framework consisting of a bridge unit (chapter IV), a path planner (chapter V), a control unit in the field (chapter VI), an artificial intelligence data

processing unit (chapter VII), and a 3D processing unit (chapter VIII). The code is available at github [5], [6], [7], [8].<sup>4</sup>

## II. STATE OF THE ART

The first stage of the mission workflow is the planner. Usually, planners for a single UAV mission run on tablets and smart phones, e.g., DJI go, Litchi or Pix4D [9]. The commercial market leader DJI offers a mobile SDK [10], but does not support the Micro Air Vehicle Link (Mavlink) protocol [11]. An advanced planner for multiple platforms (Windows, Linux, Android), i.e. QGroundControl (QGC) [12], supports the Mavlink protocol but not the DJI SDK. In 2018, the Rosetta Drone project [13] created a Mavlink wrapper for the DJI SDK. Other market participants like microdrones [14] or ASCtec [15] support only their proprietary software. The ardupilot project supports an experimental beta mission planner limited to swarming or formation-flying with multiple UAVs [16] as well as QGC and APM planner 2.0 [17]. Both implement a more sophisticated swarming / multiple UAV control.

Intelligent online or offline evaluation of the UAVs sensor data – mainly the video / image data – can be done with several deep learning frameworks, e.g., tensorflow, pytorch, keras, Caffe, Torch, MXNet, Theano, scikit learn [18].

Online or offline 3D reconstruction / mapping (Aerial photogrammetry) from multiple images can be done with, e.g., VisualSFM, COLMAP, OpenMVS, OpenDroneMap, MVE, DroneDeploy, Pix4D Mapper, AutoDesk, Agisoft PhotoScan [19], [20], [21] or online with, e.g., LDSO, ORB2 Slam or SVO / REMODE [22], [23], [24].

Some other European projects address the deployment of (teams of) UGVs and UAVs in various disaster response scenarios. ICARUS [25] and DARIUS [26] target the development of robotic tools that can assist during disaster response operations, focusing on autonomy. SHERPA [27] is focused on the development of ground and aerial robots to support human-robot team response in an alpine scenario.

## III. COMMON UAV FRAMEWORK

We created a common UAV framework to provide a general communication interface for UAVs of different manufacturers to the system (fig. 1). On the core system, a process called UAV bridge is providing the interface for the UAVs. On the UAVs' side, there are hard and software components, specific to the manufacturer or model, that communicate with the UAV bridge.

\*This work was initially funded by EU-FP7-ICT grant No. 609763 (TRADR <http://www.tradr-project.eu/>) and the current development by the Federal Ministry of Education and Research (BMBF) under grant number 13N14860 (A-DRZ <https://rettungsrobotik.de/>).

<sup>1</sup>University of Applied Science, Gelsenkirchen, <sup>2</sup>retired, former at Fraunhofer Institut für Intelligente Analyse- und Informationssysteme IAIS, Sankt Augustin, <sup>3</sup>Smart Robotic Systems GmbH, Dortmund.

corresponding author: [hartmut.surmann@w-hs.de](mailto:hartmut.surmann@w-hs.de)

<sup>4</sup>Video: <https://www.youtube.com/watch?v=WeAV07saTWM>

# UAV Integration - Overview

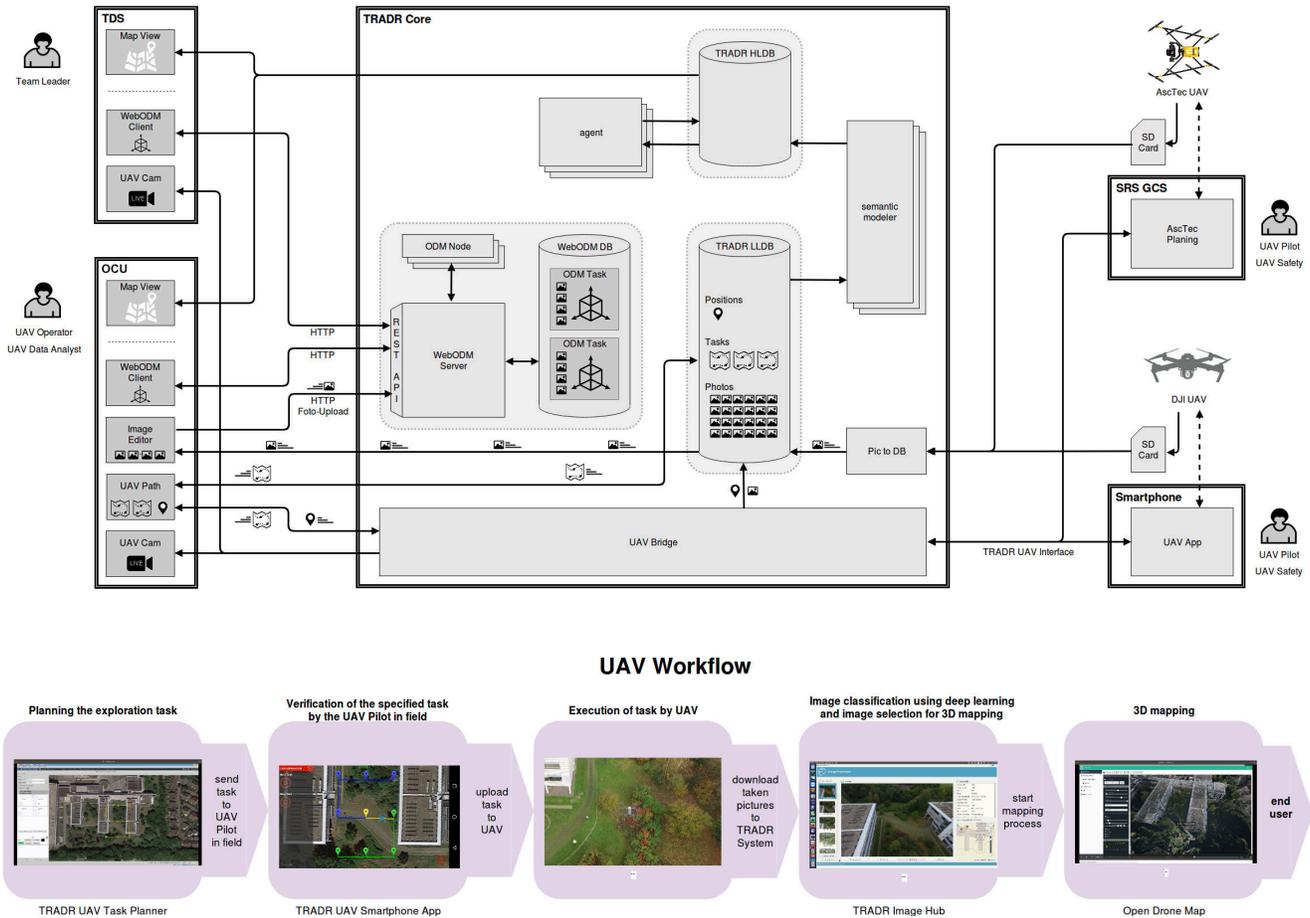


Fig. 1. UAV integration - overview

We included some rather inexpensive commercial off-the-shelf UAVs in the system like the DJI Mavic Pro, DJI Phantom 3, and DJI Phantom 4 Pro. These DJI UAVs communicate with a smartphone that the UAV pilot uses when moving in the field. The smartphone app is connected to precisely one UAV and controls this.

In the case of AscTec Neo and Falcon 8, the UAV communicates with a Ground Control Station (GCS) developed by the company Smart Robotic Systems (SRS). The GCS is part of the SKIDER [28] platform provided for the project by SRS. It is capable of managing several AscTec UAVs simultaneously.

The GCS as well as the smartphone apps controlling DJI UAVs are attached to the system network. The smartphones are connected wirelessly to the system, while the GCS is connected via an ethernet link and maintains its own wireless communication links to UAVs and pilots.

The communication between the UAV bridge and the control equipment of the UAVs is based on a dedicated application protocol, called UAV interface, which allows to

connect UAVs of different manufacturers to the system. The UAV interface defines all necessary actions and data types to interact with a UAV. The interactions include

- Registration
- Tasking
- Tracking
- Sensor data transport

The implementation of the interface is based on Google's Protocol Buffers (protobuf) library. This approach allows the use of the interface with many platforms and programming languages. As a result, there is an excellent perspective for further integration of UAVs of other manufacturers.

At the transport layer, the UAV interface can use TCP or UDP. The interface protocol defines which types of messages are sent via TCP and which are sent via UDP, based on latency-reliability considerations, e.g., messages that trigger an action are sent via TCP, video streams are sent via UDP.

The protocol defines an initial registration process, which is necessary for the system to get knowledge about the

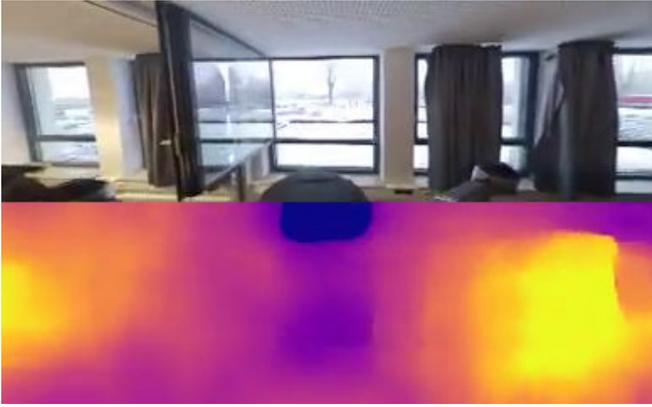


Fig. 2. Depth estimation for a live monocular videostream while flying with a Mavic Pro through an open window [29], [30]

characteristics of the deployable UAVs. Each smartphone app registers the UAV that is controlled by it to the UAV bridge. Likewise, the GCS registers the UAVs it manages. The UAV bridge continuously informs the components of the Operator Control Unit (OCU) about the registered UAVs. Thus the persons in the control room are always aware of which UAVs are available for the mission.

Each registered UAV transmits its status to the system whenever it is changed. The protocol of the UAV interface specifies how this should happen. The UAV bridge forwards this information to the components of the OCU. Thus, it is possible, e.g., to display the position of the UAVs in the maps of the OCU components.

A crucial part of the UAV interface is the capability to transfer live images from the UAVs' cameras. It defines how a component of the OCU can request the live stream of a particular UAV and how the selected UAV transmits the stream data over the interface. The live image is transmitted in an H264 coded version. The telemetry data are also transmitted in the data packets, so that these data are linked to the video frames. Only the UAVs, which were requested to transmit a live stream, will send video data to the system. This saves bandwidth on the network infrastructure and computation load on the decoding machine. The core system is responsible for the decoding of the H264 encoded streams and the distribution of the decoded streams to the OCUs. Fig. 2 shows an example while flying through an open window.

The UAV interface specifies how the UAV path planner can submit defined exploration tasks to a specific UAV or its pilot. An exploration task is described by a sequence of waypoints, which are served by the UAV in a defined order. At each waypoint, a sequence of actions can be specified that are executed by the UAV. An action could be, e.g., capturing of an image in a specific direction with defined camera settings. Furthermore, it also describes how feedback is sent to the system while the UAV executes such a task. The UAV bridge passes this feedback information so that it can be viewed in the OCU.

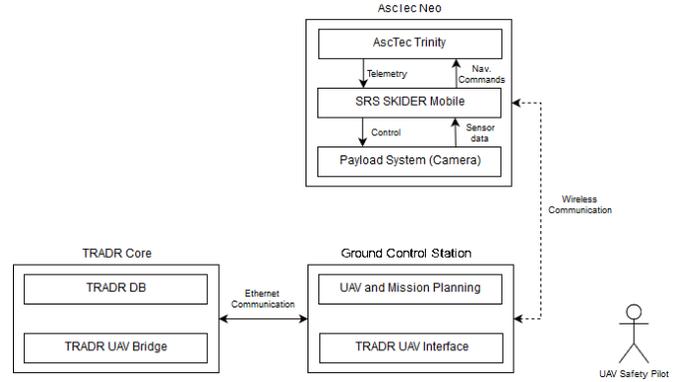


Fig. 3. UAV Integration - AscTec Neo

#### IV. INTEGRATION OF UAVS > 2KG.

The integration of large UAVs is demonstrated with two AscTec platforms: Neo and Falcon 8. Therefore, the system was extended with the SRS SKIDER [28] platform. Due to the different APIs and software versions of Neo and Falcon 8, the integration approach is quite different and thus described for each platform specifically.

##### A. Integration of AscTec Neo

The AscTec Neo is equipped with the SRS SKIDER Mobile system. SKIDER Mobile is able to send navigation commands to the AscTec Trinity flight controller and to process sensor data during flight. The AscTec Trinity SDK is completely different from the previously available AscTec ACI interface that is used for the Falcon 8. It also only provides few control and navigation commands, which implies that SKIDER Mobile must perform flight control functionalities like wind and drift compensation. On request by the operator, SKIDER Mobile captures images or videos from the camera, adds geoinformation to the image data and transmits the data to the GCS.

The SKIDER communication system ensures reliable control and payload communication to the GCS. Besides UAV and mission planning modules SKIDER Base implements the UAV Interface to allow control and data message exchange with the core system. The architecture to integrate AscTec Neo is depicted in fig.3.

##### B. Integration of AscTec Falcon 8

The AscTec Falcon 8 is a highly integrated flight platform. In order to integrate it into the system and to enable navigation control by operators, an additional component at the AscTec remote control is necessary. There is currently no option to include a companion computer or any additional component into the flight platform itself. A version of the ACI to interact with the AscTec Trinity flight controller is accessible at the remote control. This also means that the data communication with the UAV for video and payload cannot be modified. The integration architecture is depicted in fig. 4.

In order to forward the video and telemetry data to the system, the data has to be captured at the remote control.

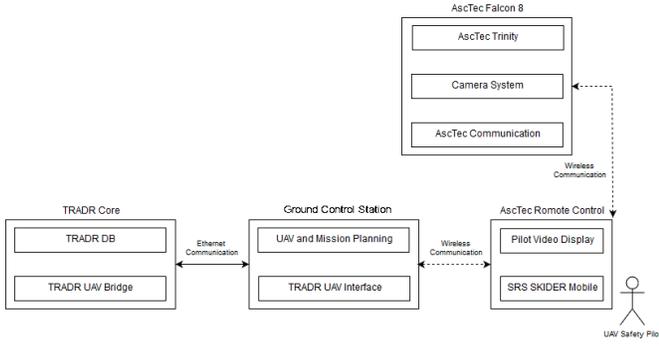


Fig. 4. UAV Integration - AscTec Falcon 8

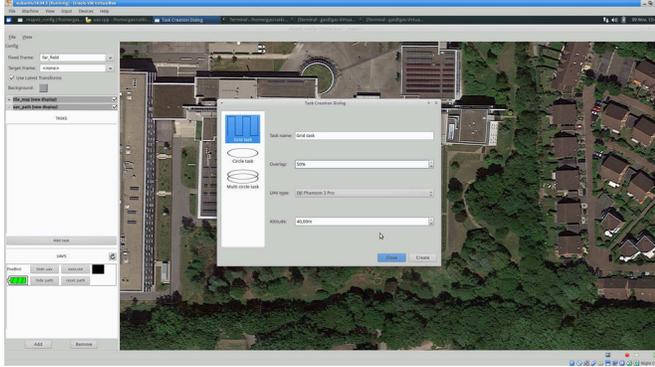


Fig. 5. Task creation dialog in UAV path planner

SRS, therefore, added an optimized SKIDER Mobile to the remote control, which is able to capture the camera’s video stream. Due to the analog transmission system on the drone, only PAL resolution is currently available. SKIDER Mobile interacts with the ACI in order to send navigation commands to the UAV. The captured data is forwarded to the GCS and delivered to the backend via the UAV interface, analogously to the Neo integration path.

## V. UAV PATH PLANNER

A path planner is implemented as a Mapviz[31] plugin. It enables the UAV operator to plan and monitor tasks executed by UAVs, which are connected via the UAV Bridge. The user can create 3 types of tasks (fig. 5):

### 1) Grid task

A grid of waypoints for the UAVs is created with this task. It is used to either create overview images or large scale point clouds representing the area of interest. Five images are taken at each waypoint: four along the X and Y-axis of the grid with a customizable camera pitch, and one image downwards. An overlap factor, which indicates the overlap between neighboring downward pictures, and the desired altitude are used to define the distance between the image positions.

### 2) Circle task

This task consists of waypoints that are arranged in a circle around a point of interest. The smaller number of images accelerates the structure from motion process for smaller regions. The number of images taken is

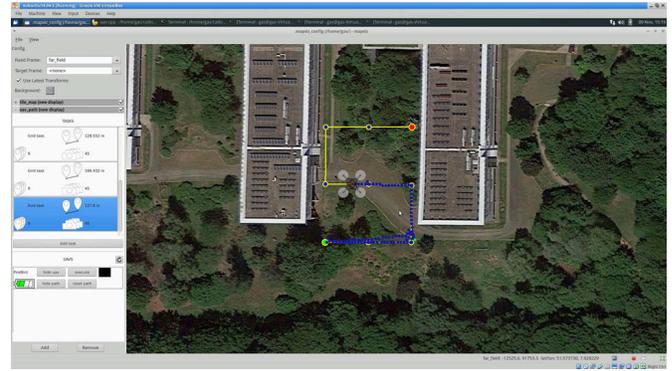


Fig. 6. Task execution monitoring in UAV path planner

calculated based on the diameter of the circle. The camera points at the ground level point under the centre of the circle when taking each image.

### 3) Multi circle task

This task consists of two circles at different altitudes in order to improve the structure from motion results. Additionally, it is possible to construct multiple neighboring pairs of circles.

A newly created task is persistently saved in the database and can be selected by the user; therefore tasks can be repeated over multiple sorties. The user can see the task name, length in meter, number of waypoints, and number of images. All waypoints of a selected task are displayed on the map. UAVs connected to the UAV bridge are listed with name, battery level, and UI elements. Optionally, UAVs and flown trajectories can be visualized on the map, and additional telemetry data can be overlaid. The UAV operator can send a selected task to a connected UAV and monitor its execution after it was approved by the pilot (fig. 6).

## VI. UAV CONTROL IN THE FIELD

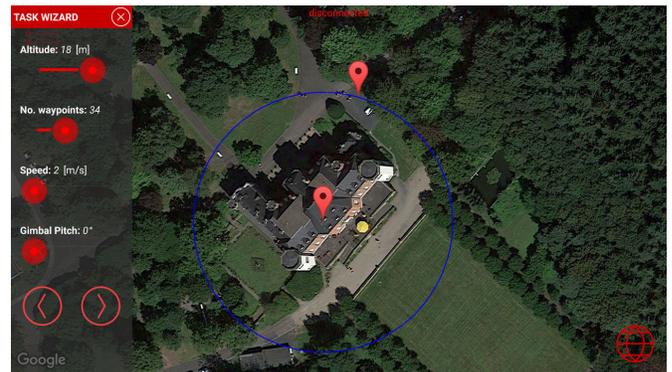


Fig. 7. Creation of a circle task in the app

The UAVs will be controlled by an operator in the field, because only a UAV pilot in the field can assess the situation well enough to control the UAV safely. This pilot stays in line-of-sight with the flying UAV permanently, which is also necessary for legal reasons. The UAV pilot controls the UAV

manually over a standard remote control unit supplied by the manufacturer of the UAV.

The completed task can be uploaded to the UAV (fig. 10). The UAV always stops its movement before it takes an image in order to prevent image degradation from motion blur, especially in low-light conditions. If the smartphone app is connected to the system, the UAV's telemetry data (such as position, orientation, speed, battery level), as well as the live image from the UAV's camera, is continuously transmitted to the control room. The connection to the system also allows the UAV pilot in the field to receive tasks from the UAV operator stationed in the control room. In this case, the app notifies the UAV pilot that it has received a new task. The UAV pilot has then to ensure that it is possible to perform the requested task safely by reviewing the included trajectories and actions. In case the UAV pilot deems the task to be unsafe, he can reject it, which is then signaled to the UAV operator in the control room. Otherwise, he can accept the task, and after that, he can transmit the task data to the UAV and start the execution. While the UAV is executing the task, the UAV pilot monitors the process and can intervene at any time, cancel the automatic flight and manually control the UAV again. Furthermore, during the execution of the task, the feedback data is continuously sent back to the UAV operator in the control room, who requested the execution of the task. Thus, the UAV operator can track the progress of his task in real time.

In the case of DJI, this remote control unit is connected to a smartphone that runs the UAV app developed for the TRADR project. It fulfills two tasks:

- 1) The UAV can execute previously defined tasks automatically.
- 2) Information is transferred between the UAV and its pilot and the rest of the system.

The app displays a map showing the UAV's current position and orientation as well as the live image of the UAV camera. It is easy to create standardized tasks that are needed in the rescue operation. This is very useful in situations where the UAV mission starts before the entire system is up and running. As soon as the system is operational, the previously collected data can be incorporated into the mission database for further processing by the UAV operator and providing benefits to the rescue services.

With the circle task (fig. 7), the UAV flies in a circle around an object in its centre, taking images from different perspectives. The pilot can specify the object to be inspected in the map. Additionally, the radius of the circle, the altitude, and the camera tilt can be adjusted as well as the number of points on the circle trajectory, where the UAV takes the images.

The area task (fig. 8) enables the pilot to define an area, over which a meander-like trajectory (fig. 9) is fitted, by placing the vertices of a polygon on the map shown by the smartphone app. The objective of this task is to achieve complete image coverage of the area, for the creation of 2D and 3D maps. To this end, the app orients the meander along a grid inside the area, and the UAV takes five images at every

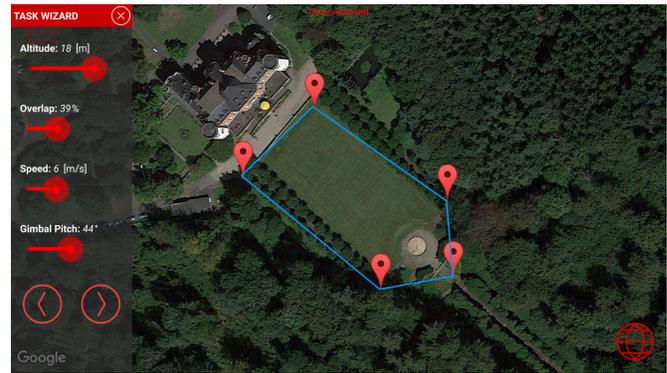


Fig. 8. Creation of an area task



Fig. 9. Generated meander like trajectory

grid intersection. For four of these images, the camera will be pointing towards the cardinal directions while tilted by a user-specified angle, and the fifth image is captured with the camera pointing to the ground. The spacing of the grid is calculated from the desired overlap between images taken from neighbouring grid intersections.

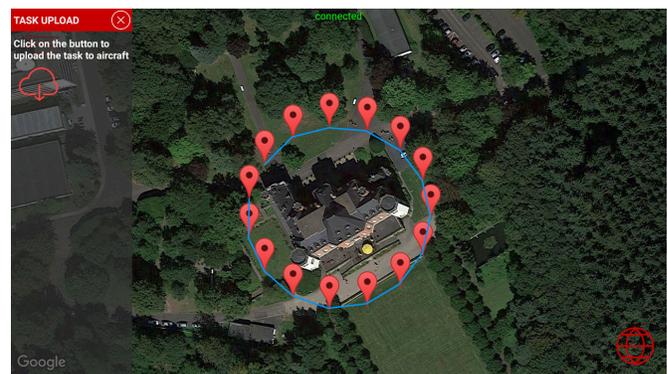


Fig. 10. Circle trajectory ready for upload to the UAV

## VII. INTELLIGENT IMAGE HUB

The UAV operator uses an intelligent image hub to analyze and evaluate the images that were created by the successful execution of a UAV task. The results are very important for the planning of further UAV tasks and the generation of 3D

models or ortho maps. The process needs much concentration and comprehension. The intelligent image hub improves this situation and supports the UAV operator to emphasize and to present crucial information.

The integrated semantic modeler in the back end of the system marks relevant objects, e.g. cars, in the images and calculates the GPS coordinates of the recognized objects if necessary. This information is stored in the database. The object recognition was implemented by a pre-trained convolutional neural network, trained on the COCO dataset[32]. To account for the perspective of UAV footage, further training was conducted with the CARPK and PUCPR+ datasets[33] and our images from the TRADR project. The utilized network is the FasterRCNN Inception v2, which was released as part of the TensorFlow detection model zoo[34].

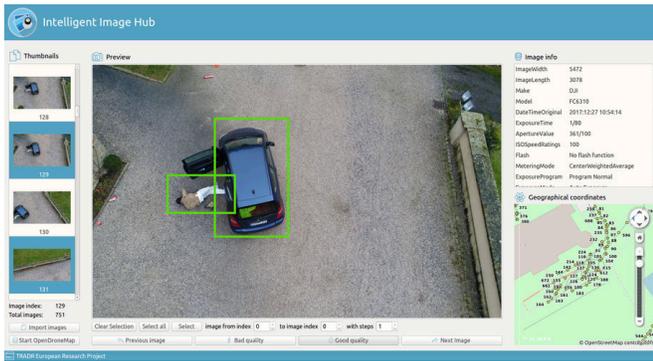


Fig. 11. Intelligent image hub for aerial images: visualisation, object annotation and object detection

In the front end of the system, the UAV operator uses the intelligent image hub to make a choice from the set of images that were recorded by the UAV on a storage media (fig. 11). During this process, the images are imported into the Database and evaluated by the semantic modeler (fig. 12).



Fig. 12. Example of neural scene classifier. The image was taken in the Dortmund firehouse of the local Fire Service college. We simulated a ceiling fire where the people move to the bottom. A pre-trained graph convolutional network for skeleton-based action recognition [29] classified the scene as feeding birds because of the absence of training material

In order to ensure a good overview of the data, the OCU displays the trajectory of the UAV, images of the UAV task,



Fig. 13. 3D modell view in WebODM



Fig. 14. Orthophoto view in WebODM

their EXIF header information, and results of the semantic modeler. The UAV operator can evaluate the results of the semantic modeler, adjust them if necessary, and annotate relevant objects in images.

Images selected in the intelligent image hub can be passed to Open Drone Map to start the process of generating 3D models or ortho maps.

## VIII. OPEN DRONE MAP (ODM)

ODM[20] is an open source toolkit for processing aerial drone images; it generates 3D models (fig. 13) and ortho photos (fig. 14) from UAV images. We included this tool in the framework as an easy-to-use pipeline for the end users to perform UAV-related postprocessing: create point clouds, ortho photos, and to make some deeper analysis. The complete pipeline is splitted into several components, all capsuled in Docker[35] containers.

First, the user has to prepare the images with the intelligent image hub to ensure that they can be automatically processed with ODM. The images are then uploaded to the API of the WebODM container; the container creates a new processing task automatically afterward. The processing task will be forwarded to the node-Opendronemap container and processed internally. When finished, execution time depending on the resolution and number of images, the node-Opendronemap returns the results to the web app WebODM.

We extended WebODM with the following components:

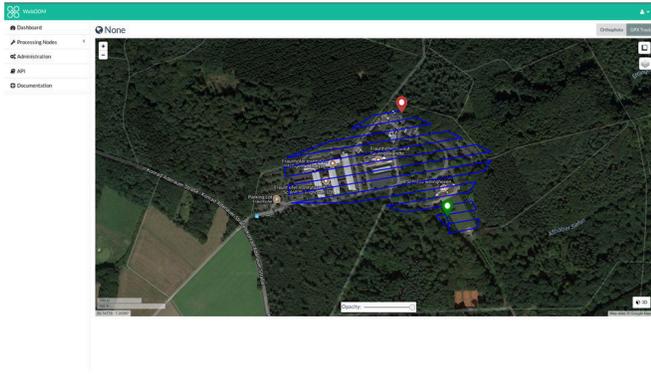


Fig. 15. Trajectory in WebODM

- integration of a reconstruction tool to view the original camera positions
- integration of the UAV trajectory in the map view, based on EXIF-headers of the images (fig. 15)

The additional web app WebODM acts as a GUI for the ODM pipeline and lets the user process images manually or lets him view the processed image sets. The 2D mapview plugin enables the end user to use several overlays, the UAV trajectory, a geo-referenced ortho photo (top view of the 3D model), elevation model, or several other maps distributed by a map server; furthermore, there is also a tool for distance measurement available. With the 3D viewer, it is possible to view point clouds and many different parameters (e.g. point size, texturing) can be adjusted. It is also possible to do measurements tasks in 3D. Finally, we have integrated a reconstruction tool, which allows the end user to navigate between the several camera positions, in a style similar to google street view.

## IX. CONCLUSIONS

This paper presented a framework for the integration of UAVs in Search and Rescue missions, which consists of a UAV task planner, heterogeneous UAVs, a bridge to these UAVs, an intelligent image hub, and a 3D point cloud generator. Tests in several TRADR missions show the potential and feasibility of the approach and also gives several feed back from rescues for future research. Tests and evaluations in several TRADR missions show the potential and feasibility of the approach and also gives valueable feed back from rescuers for future research. So, needless to say that much work remains to be done in the current project A-DRZ (Startup of the German Rescue Robotics Center).

Future work will integrate further UAVs, extend the intelligent image hub with more neural (deep learning) classifiers and add a Mavlink implementation [11] to the UAV app and the UAV bridge so that further planning tools could be integrated, e.g., QGroundControl [12].

## REFERENCES

[1] European Aviation Safety Agency, <https://www.easa.europa.eu/>.

- [2] G. Kruijff, I. Kruijff-Korbayová, S. Keshavdas, B. Larochelle, M. Janíček, F. Colas, M. Liu, F. Pomerleau, R. Siegwart, M. Neerincx, R. Looije, N. Smets, T. Mioch, J. van Diggelen, F. Pirri, M. Gianni, F. Ferri, M. Menna, R. Worst, T. Linder, V. Tretyakov, H. Surmann, T. Svoboda, M. Reinštejn, K. Zimmermann, T. Petříček, and V. Hlaváč, "Designing, developing, and deploying systems to support human-robot teams in disaster response," *Advanced Robotics*, vol. 28, no. 23, pp. 1547–1570, 2014.
- [3] I. Kruijff-Korbayová, F. Colas, M. Gianni, F. Pirri, J. de Greeff, K. Hindriks, M. Neerincx, P. Ögren, T. Svoboda, and R. Worst, "TRADR Project: Long-Term Human-Robot Teaming for Robot Assisted Disaster Response," *KI - Künstliche Intelligenz*, pp. 1–9, 2015.
- [4] I. Kruijff-Korbayová, L. Freda, M. Gianni, V. Ntouskos, V. Hlav, V. Kubelka, E. Zimmermann, H. Surmann, K. Dulic, W. Rottner, and E. Gissi, "Deployment of ground and aerial robots in earthquake-struck amatrice in italy (brief report)," in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Oct 2016, pp. 278–279.
- [5] <https://github.com/RoblabWh/tradr-uav-ros-interface>.
- [6] <https://github.com/RoblabWh/tradr-uav-fpv>.
- [7] <https://github.com/RoblabWh/tradr-uav-bridge>.
- [8] <https://github.com/RoblabWh/tradr-uav-app>.
- [9] <https://play.google.com/store>.
- [10] <https://developer.dji.com/mobile-sdk>.
- [11] <https://github.com/mavlink/mavlink>.
- [12] <http://qgroundcontrol.com/>.
- [13] <https://github.com/diux-dev/rosettdrone>.
- [14] <https://www.microdrones.com/en/integrated-systems>.
- [15] <http://www.asctec.de/>.
- [16] <http://ardupilot.org/planner/docs/swarming.html>.
- [17] <http://ardupilot.org/planner2>.
- [18] A. Kothari, "Top 9 frameworks in the world of artificial intelligence," <https://geekflare.com/ai-frameworks>.
- [19] M. von Uebel, "16 best photogrammetry software tools in 2019," <https://all3dp.com/1/best-photogrammetry-software/>.
- [20] P. Toffanin, <http://opendronemap.org/>.
- [21] S. Fuhrmann, F. Langguth, and M. Goesele, "Mve: A multi-view reconstruction environment," in *Proceedings of the Eurographics Workshop on Graphics and Cultural Heritage*, ser. GCH '14. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2014, pp. 11–18.
- [22] X. Gao, R. Wang, N. Demmel, and D. Cremers, "Ldso: Direct sparse odometry with loop closure," in *iros*, October 2018.
- [23] M. J. M. M. Mur-Artal, Raúl and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [24] M. Pizzoli, C. Forster, and D. Scaramuzza, "REMODE: Probabilistic, monocular dense reconstruction in real time," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [25] G. De Cubber, D. Doroftei, D. Serrano, K. Chintamani, R. Sabino, and S. Ourevitch, "The eu-icarus project: Developing assistive robotic tools for search and rescue operations," in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Oct 2013, pp. 1–4.
- [26] P. Chrobocinski, E. Makri, N. Zotos, C. Stergiopoulos, and G. Bogdos, "Darius project: Deployable sar integrated chain with unmanned systems," in *2012 International Conference on Telecommunications and Multimedia (TEMU)*, July 2012, pp. 220–226.
- [27] L. Marconi, S. Leutenegger, S. Lynen, M. Burri, R. Naldi, and C. Melchiorri, "Ground and aerial robots as an aid to alpine search and rescue: Initial sherpa outcomes," in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Oct 2013, pp. 1–2.
- [28] <http://www.skider.de>.
- [29] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," *CoRR*, vol. abs/1801.07455, 2018.
- [30] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," *CoRR*, vol. abs/1811.06152, 2018.
- [31] <https://github.com/swri-robotics/mapviz>.
- [32] <http://cocodataset.org/>.
- [33] <https://lafi.github.io/LPN/>.
- [34] <https://www.tensorflow.org/>.
- [35] <https://www.docker.com/>.