Small commercial UAVs for indoor Search and Rescue Missions

Hartmut Surmann[1], Tiffany Kaiser[1], Artur Leinweber[1], Gerhard Senkowski[1], Dominik Slomma [1], Marc Thurow[1]

***Abstract*— This technical report is about the architecture and integration of very small commercial UAVs ($<$ 40 cm diagonal) in indoor Search and Rescue missions. One UAV is manually controlled by only one single human operator delivering live video streams and image series for later 3D scene modelling and inspection. In order to assist the operator who has to simultaneously observe the environment and navigate through it we use multiple deep neural networks to provide guided autonomy, automatic object detection and classification and local 3D scene modelling. Our methods help to reduce the cognitive load of the operator. We describe a framework for quick integration of new methods from the field of Deep Learning, enabling for rapid evaluation in real scenarios, including the interaction of methods.**

***Index Terms*— Search and Rescue Robots, Unmanned Aerial Vehicles, Artificial Intelligence, Deep Learning, Autonomous Robots**

I. Introduction

On August 24th, 2016 an earthquake of magnitude 6.2 hit central Italy [1]. One week later, on Thursday, September 1st a team of the EU project TRADR deployed three UAVs in Amatrice, Italy, to assist in the post-earthquake response. The team was asked to provide textured 3D models of two churches, San Francesco and Sant'Agostino, both in a state of partial collapse. The models were used to plan the building support, to prevent further destruction, and to preserve the national heritage monuments from the 14th century, as well as to protect the rescuers [2]. To our knowledge, it was the first time that the outcome of a mission depended on the UAVs capabilities to enter partially collapsed buildings through broken windows or holes in roofs. The buildings were entered successfully by a DJI Phantom 3 ($\sim$ 1.3 kg, $\sim$ 60 cm diagonal, 4 rotors). The two other UAVs (AscTec Falcon 8, $\sim$ 120 cm, 8 rotors) were too large to enter the buildings but provided an overview from outside the churches. From these missions we have learned several lessons:

1) UAVs are needed for USAR missions, in which damaged buildings are to be entered, which are inaccessible for humans or ground robots. In the above scenarios UAVs provide a quick scene overview given a live video stream from their onboard cameras. With series of Images made from inside the buildings a 3D model was later provided for further mission planning. No humans were put in danger.

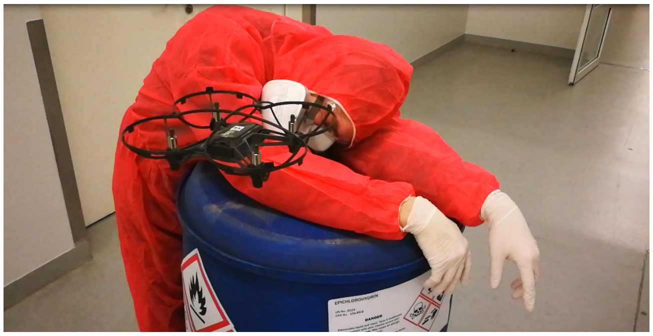

Fig. 1. Example of indoor task. A small UAV entered a building and detects an accident. The UAV is semi autonomous and remote controlled by an operator and a neural network agent (assisted autonomy). The camera images are sent to the outside operator and the AI agent. The images are processed outside and control commands are sent back to the UAV. A stable radio link connection is necessary.

2) We had great success with small UAVs mainly used as mobile sensor platform (with the camera being the most important one) with otherwise limited computation capabilities. It was a bit lucky, however, that we were indeed able to enter the buildings, since it is totally possible that the next mission requires even smaller UAVs. As such the size of applied UAVs must shrink.
3) We have experienced first-hand that the cognitive load in teleoperation mode is very taxing for a single UAV operator as he needs to focus on the navigation while he simultaneously must keep track of the environment with its dangers. This is even more imminent in stressful missions. For this reason these tasks are often split to a pair of individuals, but even then, they remain difficult to handle [3], [4]. As such, assistive techniques are needed, e.g from the field of AI.

Nowadays UAVs are available on the free market which are cheaper and smaller than the DJI Phantom 3 ($<$ 0.5 kg, $<$ 30 cm vs. $\sim$ 1.3 kg, $\sim$ 60 cm) while still being comparable in terms of functionality (see section IV for details). These UAVs provide a live video stream and Images of high resolution, but, in order to save weight and power, active distance measuring sensors aren't available. Compared to most outdoor scenarios the exploration and navigation in indoor environments quickly becomes more demanding, because of the absence of GNSS for localisation and a potentially huge number of obstacles which often are in close proximity to the UAV. Things get even harder when no 3D sensors are available, since any localisation then must rely on pho-

This work was funded by the Federal Ministry of Education and Research (BMBF) under grant number 13N14860 and 01|S19060C (A-DRZ https://rettungsrobotik.de/ and AIA https://www.aiarena.de).

[1] University of Applied Science, Gelsenkirchen.
corresponding author: `hartmut.surmann@w-hs.de`

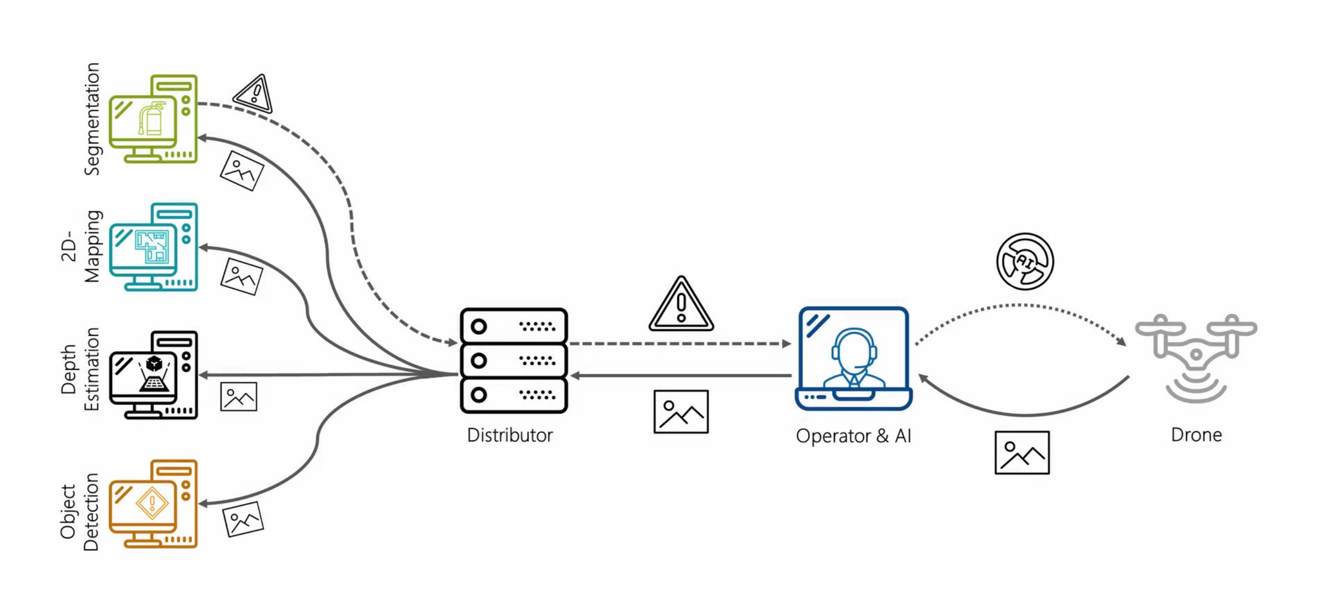

Fig. 2. Architecture for small UAVs in SAR missions. It consists of the small UAVs, a remote operator station and several computing nodes for the AI algorithms. The camera images from the drone are distributed to the AI computing nodes and the detected objects, point clouds, maps and control commands are sent back to the operator and the UAV. This approach needs a connection from the operator to the UAV.

togrammetric methods which are computationally expensive while often delivering only sparse structural information of the environment. In fact, the autonomous control of a mobile robot in every indoor scene under all circumstances lies beyond the current state of the art. However, when controlled in teleoperation mode, a human operator has to estimate depth and distances in 3D by himself, which is a tedious task with a high cognitive load. Solving complex tasks with Computer Vision, given limited resources for computation, has become a typically use case for Deep Learning. Being one of the most active research topics of robotics many methods from deep learning are still experimental and the current state of the art can change quickly. As a side effect, most methods are far from beeing optimized for target platforms. In the context of indoor rescue robotics there is only little experience in the application of methods based on Deep Learning and what kind of problems can and should be solved. As such we assume that we benefit from a system to quickly test hypotheses on live data from one or multiple UAVs. Considering the fast progress in drone technology and the uncertainty about the concrete model or the number of models to use while simultaneously testing and integrating a potential multitude of computationally demanding neural nets, we think on a system with the following properties:

1) There is a method to uniformly register and command a multitude of small and cheap, commercial off-the-shelf UAVs in one system, considering that different vendors offer different SDKs and Interfaces. Flying indoor, we refer to small UAVs equipped with cameras.
2) So far, UAVs should be used in a similar fashion like in Armatrice: One human operator controls one UAV which provides a live video stream and series of Images from the indoor environment. AI based methods might help but not fully substitute the operator for the control task under all circumstances.
3) In order to consider the latest state of the art from the very active research fields of deep learning and AI and to evaluate their potential for inddor USAR missions, quick deployment, integration and testing of new and computationally expansive methods is realized with a strong backend at hand and a permanent network connection to the UAVs.
4) There is a basic scenario with application to real USAR missions, which is derived from our observations with TRADR or other projects. The application of certain methods from the current state of the art of deep learning can be evaluated.

With this report we describe our current progress in the development of such a system, starting from a somewhat basic, fictional scenario so far: One UAV has been maneuvered manually by a single operator through a building until it reaches a corridor of a larger corridors system. The UAV must now be maneuvered through the corridor, which is tedious work. At some point though, there is an injured person who is in acute danger, as he appears to have direct contact to a possibly damaged container filled with hazardous substances (see fig. 1). We considered several branches of current research topics in deep learning to integrate them in our system. We wanted to support the operator on several levels: We considered an autonomous flight through corridors to be a simpler subproblem of general autonomous indoor flight as no complex decisions regarding wayfinding are to be made and in most cases, ways are free from obstacles. One of our hypotheses was that this task can be solved by a neural net derived from existing methods for the autonomous navigation of forest trails. Another one was that segmentation and classification could be applied with the help of deep nets. Finally we hypothesized that DNN based depth estimation of the current image input could be used for collision avoidance and local 3D modelling. The latter one could maybe help to increase the spatial awareness of the operator. The outcome on the AI side of the system is the following:

1) We have implemented a fast DNN which controls the UAV through corridors autonomously. This provides the operator with the opportunity to focus solely on the scene.
2) We evaluated several methods for Object detection, classification and segmentation, e.g. for the detection of humans. One essential task for rescuers in collapsed buildings is the search for trapped and injured humans. Autonomous corridor flight might lead to overlooked humans in acute danger so a person detection algorithm seems vital. Warning signs at doors, walls or containers might give hints regarding the nature of the catastrophe or what kind of upcoming catastrophe could still happen. Automatic detection and classification also reduces the cognitive load of the operator.
3) We tested a DNN for direct depth estimation on 2D image input for the sake of collision avoidance and local 3D modelling to improve spatial awareness and to ease manual navigation. We observed that current methods are not reliable enough, e.g. to ensure collision avoidance. We took a current method and optimized the training with respect to the special scene structure of indoor corridors. We think that considering the specialities of the expected scene structure can lead to more reliable applications in practice.

The paper is structured as follows. In the next chapter we describe the state of the art regarding the application and integration of small UAVs on USAR missions and related topics. In chapter three we provide an overview over some relevant UAVs available on the free market. In chapter four we describe our current progress in the development of a distributed system architecture for quick integration of small UAVs and new AI/deep learning based methods. The final chapters deal with the implementation, training and evaluation of our individual solutions in real corridor scenes.

II. Related work

The integration and application of UAVs in Search and Rescue missions has recently been researched in several projects such as TRADR, EffFeu and Eins3D with different priorities. In TRADR (Long-Term Human-Robot Teaming for Disaster Response) [5] [6] multiple UAVs and UGVs with differing capabilities delivered different data which was sent to a central system where that data was fused, processed, persistently saved and finally presented to USAR forces. AI was present at several levels of the project, but the main focus was more on the representation, presentation and use of fused data. Scenarios referred either to outdoor missions, indoor missions or mixed missions with compositions of UAVs and UGVs. Both UAVs and UGVs possessed autonomous modes which could optionally be activated by a single responsible operator in outdoor scenarios. For UGVs autonomous operation modes were also available in indoor environments, where UAVs were only flown manually.

The research project EffFeu (Efficient Operation of Unmanned Aerial Vehicle for Industrial Firefighters) [3] focused on the integration of UAVs in the work of industrial firefighters. It provided an autonomous mission-guided control on goal-oriented high-level tasks, such as the search for humans, for both, indoor and outdoor environments. For outdoor environments GNSS was used for the localisation, while in GNSS denied areas (e.g. indoor areas) this was performed by ORB-SLAM2 with RGB-D sensors in combination with alignment with prior known maps (which are available to industrial firefighters in practice). The project also applied deep-learning based object recognition of relevant objects, similar to our approach. However, unlike ours, the proposed system was limited to simulated scenarios and UAVs and did not cover the selection of specific devices and their integration into the system. This includes the absence of considerations such as size, weight and the stability of the UAVs trajectory. Furthermore, while an AI-based object detection algorithm was implemented, there was no focus on quick integration and evaluation of experimental methods of this field.

The project Eins3D (Luftbasierte Einsatzumgebungsaufklärung in 3D) [7] was aimed at the development of a single drone in combination with a real time capable 3D mapping for the purpose of delivering an overview of the situation to USAR forces. The utilized UAV was a DJI S1000+ which is too large for most indoor scenarios. AI methods had not been part of Eins3D.

Croon and De Wagter identified requirements for the autonomous navigation of UAVs in indoor environments [8], also providing an overview of the current state of the art. According to the authors, flying indoor with UAVs faces several challenges:

1) Low traversability due to close spatial proximity and high collision probability.
2) Usually only 2D cameras are available, but no active 3D sensors, due to size and power constraints. The reconstruction of 3D information needs more computational power than small UAVs can provide.
3) UAVs are often not perfectly stable on their current position, which can cause collisions. The drift can not be balanced by accurate localisation onboard, see previous point.
4) The authors recommended the usage and exploration of AI methods to solve complex tasks, e.g. depth estimation with deep neural networks and navigation with deep reinforcement learning. They remarked that some impressive progress has been made regarding autonomous indoor navigation so far, but that there is still no general and reliable autonomous navigation. The authors suggest that open challenges for specific scenarios should next be investigated.

III. Architecture

Our current architecture, as shown in fig. 2, consists of the following main parts:

1) A lightweight (<0.5 kg) consumer-grade UAV with a monocular camera.
2) A mobile control station (e.g. a laptop) with a guaranteed network connection to the UAV, which provides the UAV with the most critical control commands.
3) Multiple computation nodes (e.g. with PCs) with or without GPU support for complex data processing tasks from the AI domain (e.g. DNNs).
4) A distributor to establish the communication between the UAV and other computation nodes. Both unidirectional and bidirectional transfer modes are supported. Meanwhile, the control station might also be a target for some of the backend nodes and can also be registered to the distributor.

The mobile control station is used by the operator to control and supervise the UAV in teleoperation mode, but once a corridor is entered, the control can be switched to AI-assisted autonomous flight which allows the operator to shift focus from flight control to mission execution. Furthermore, the mobile control station forwards image data from the UAV to computation nodes in the network and receives processed data. The topological proximity of the control station to the UAV reduces latency and increases network stability for some of the most time-critical parts of the UAV control. The distribution of data is implemented as a separate process (”The distributor”) which can also be deployed on a second computer. This is optional but it can help to reduce the load on the control station if this is required. We provide an easy-to-use interface to register new processing tasks in the network. Data processing may include computationally expensive methods, e.g. we make use of recent deep convolutional neural networks (DCNN). This requires powerful CPUs and graphics cards on the computation nodes. Depending on the load and the available resources, a node may even run multiple tasks at once. Besides the previously mentioned AI-based tasks which we have implemented so far, other methods can quickly be added or existing tasks can easily be exchanged due to the simplicity of our registration process. Thus, while being a very basic architecture, it sufficiently supports the integration and evaluation of complex systems of various state of the art methods in real applications.

IV. UAVs

The following devices are exemplary for commercially available UAVs in the $< 0.5\,\text{kg}$, $< 40\,\text{cm}$ range. The Ryze Tech Tello EDU is a small ($98 \times 92.5 \times 41\,\text{mm}$ ($L \times W \times H$)), light ($0.08\,\text{kg}$) and inexpensive (155 Euro) UAV designed for educational purposes[1]. It carries a fixed camera capable of 720p video and is stabilized through an onboard vision positioning system. As part of DJI’s lineup of UAVs both the Spark and the Mavic Mini are light ($0.3\,\text{kg}$ and $0.249\,\text{kg}$) and small ($143 \times 143 \times 55\,\text{mm}$ and $160 \times 202 \times 55\,\text{mm}$ ($L \times W \times H$)) and are therefore suitable for indoor operations [2]. These UAVs are equipped to stream 1080p or 2.7K video streams from their gimbal-mounted cameras. The Tello has the shortest flight time of all these UAVs of just 13 minutes. The other two UAVs can fly for up to 30 minutes. For our initial implementation, we chose the Tello because of its robustness, size, and simple interface. An additional protective cage keeps the UAV and its environment safe. This became apparent while testing experimental software in small corridors.

To communicate with the mobile control station, the Tello creates an ad-hoc WiFi network. This limits the range of the UAV to 100 meters under optimal conditions without interferences. For indoor flights, the range is reduced substantially, and it was necessary to follow the UAV with the laptop during our experiments, therefore its use is limited to development and quick testing. The Tello SDK 2.0 allows us to receive a video stream and status information from the UAV and to send control commands programmatically.

The other DJI UAVs use specialized hardware and communication protocols in their remote controls allowing for much longer flight distances (e.g. up to 2000 metres for the DJI Mavic Mini). DJI provides the DJI Mobile SDK for the DJI Spark and Mavic Mini to interface with those proprietary devices. However, to easily integrate a multitude of different UAVs, a common and open communications interface is necessary. The MAVLink protocol provides such an interface for small UAVs and libraries with MAVLink support are available for many programming languages. This gap between the DJI Mobile SDK and MAVLink is bridged by Rosetta Drone, a wrapper that allows us to communicate with DJI UAVs via MAVLink.

We will continue with the integration of DJI UAVs with Rosetta Drone as part of our contribution to the A-DRZ project. The increased range over the Ryze Tech Tello will allow us to establish the control station at a safe distance from potentially dangerous areas. The suitability of small UAVs for enclosed spaces will be evaluated further during the A-DRZ project.

V. Navigation

To relieve the UAV pilot from the mental strain of precisely controlling the UAV over a long time, we wanted to offer the ability to activate an assistive function that steers the UAV through corridors. When enabled, the UAV follows the hallway in the current direction, and the pilot has to disengage the function manually to change the route or inspect some point of interest in the environment.

For this we build on the work of Giusti et al. [9]. In their paper they describe a CNN which was trained to follow forest trails. We assume that following a forest trail and a corridor are similar tasks and that a corridor is even simpler to navigate because it does not contain the same amount of

[1] specifications from `https://www.ryzerobotics.com/de/tello-edu/specs`

[2] specifications from `https://www.dji.com/`

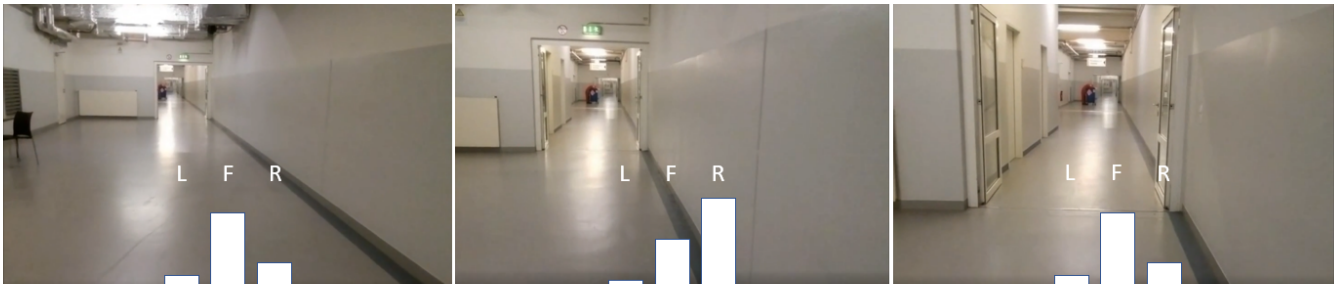

Fig. 3. Illustration of the three output classes of our neural network in different situations. The middle bar represents the velocity of the UAV directly. From the other two output values, represented by the outer bars, the turning direction is calculated by the difference between these two values.

relevant features as a forest trail. Therefore, we shortened the original architecture by removing two pairs of convolutional and max-pooling layers. Analogous to [9] we used a setup of three GoPro action cameras, one directed forward, two angled to the sides, to collect 90000 images of corridors in the building of the Westphalian University of Applied Sciences. From each image triple the outer two images were labeled as ”move left” or ”move right” respectively, the middle one with ”forward”. To encode this information numerically we used a one-hot-encoding style. All other aspects of the training were identical to that described in the work of Giusti et al. A cheap mid-range Nvidia Geforce GTX 1060 6GB was sufficient to train the network for 300 epochs or 3 hours. We then chose the epoch with the best evaluation result. The ”forward” output of the neural network is linearly mapped to the velocity of the UAV by some empirically found eta. In contrast to our training data the image input from the UAV under practical conditions exhibits situations where the neural net could find a reason to turn both left and right, e.g. if the UAV is close to a wall of a corridor and looking towards the opposite wall of the corridor. For this reason we determine the turning direction by computing the difference between the remaining two outputs (fig. 3).
With the shortened CNN we reach a control loop time of just four milliseconds, which is more than enough to fulfill the time constraint with a 30 fps video stream. As the navigation is a time critical task we run it on the control station in close proximity to the UAV to avoid additional latency. While our approach to follow corridors without collisions works fine in most cases, see our video on youtube[3], we noticed some situations where the UAV gets stuck on seemingly unimposing details in the environment and continues to alternate between turning left and right quickly.

To improve the performance of the corridor flight, we consider modifying the network so that lateral movement becomes possible. Another idea is to use multiple consecutive frames as input or to incorporate an LSTM layer to prevent the UAV from getting stuck in fast oscillating moves. Another concept worth investigating is the combination with depth estimation in a concurrent way where the network retains its quick reaction time but benefits from the additional depth information, which is updated every N frames.

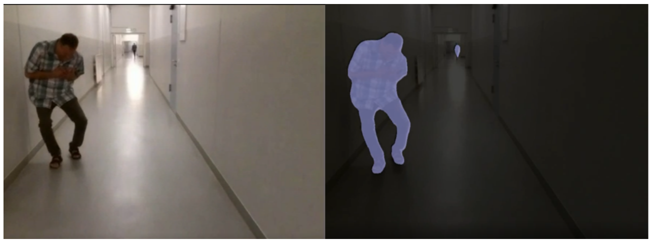

Fig. 4. Semantic segmentation of a person in acute danger in realtime using a lightweight RefineNet.

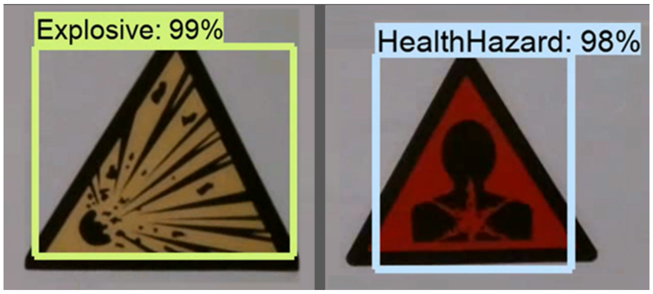

Fig. 5. Detection and classification of warnings signs in real time with R-FCN

VI. Semantic Segmentation and Object Classification

While our autonomous corridor flight system helps to reduce the cognitive load of USAR forces in stressful situations it comes with a drawback: Losing focus on the scene, certain objects of interest, e.g. containers for dangerous goods or humans in acute danger could be overlooked. For this reason a real-time capable automatic object detection and classification system is required. Such a system could either be used as part of the autonomous navigation routine itself or for a sudden switch from autonomous flight to manual control. Another major benefit from automatic object classification is its step towards automatic scene understanding. If two individuals were previously working together to control the UAV, one for the navigation, the other for scene observation, an automatic scene understanding would be a major step towards a single-handed UAV application. Our current progress in object detection and classification falls behind the other areas, so we rather focussed on quick experiments than on methodologically clean solutions. Here

[3] https://www.youtube.com/watch?v=muQAA7_2ZdU

we benefit from our rapid integration of new tasks in our system.

We rapidly integrated a lightweight RefineNet for real time semantic segmentation [10] on one of the computation nodes in the backend in order to detect and to classify important objects in our scenario, including humans, containers (as shown in the example above) and bottle-like-structures such as fire extinguisher. We trained different versions of RefineNet with varying sizes on multiple, freely available datasets for semantic segmentation including the PascalVOC dataset [11], Pascal Person Part, Pascal Context and NYU. We found that even a small-size RefineNet based on RF-LW-ResNet 50 [10], trained on Pascal VOC can show promising results for real-time segmentation of humans and fire extinguishers in corridor scenes (see fig. 4), while others, like the Pascal Person Part, were less accurate.

In order to detect and classify relevant warning signs we trained a R-FCN (Region-based Fully Convolutional Networks) [12] on two datasets with different warning signs. The first dataset contained a set of eight warning signs consistent to ISO 7010 [13] and preceeding standards, the seconded dataset included images with six types of pictograms consistent to the Globally Harmonized System of Classification and Labelling of Chemicals (GHS) [14]. For each dataset we created various perspectives of each warning sign using a homography transformation and inserted them in images of different scenes. Warning signs were scaled, rotated and skewed in the process. In total our dataset consisted of 8459 images or 7274 images with pictograms respectively. In both cases we continued the training of models which were pretrained on the COCO dataset [15]. While this allowed us to detect warning signs on UAV footage, as seen in fig. 5, we noticed some situations where the detection failed, probably due to missing edge cases in the training data.

However, there is still a lot of work for us to do regarding object classification, namely in the evaluation and the interaction with the other tasks of our system. One important feature of the A-DRZ is the integrative and cooperative work between researchers in technology and real USAR forces. In the near future we plan to identify objects essential for real USAR mission scenarios, leading to dataset creation, model design and training as well as to integrative testing in real scenarios.

VII. Mapping

The sparse point clouds of SLAM algorithms are often difficult to interpret if the user has no prior knowledge about the geometry of the underlying scene. This is mostly due to the circumstance that monocular SLAM algorithms focus on tracking and localisation and are constrained by realtime requirements.

In our mapping approach, we attempt to improve the situation by combining a popular SLAM method with semantic segmentation to create a dense floor map. With the indirect ORB-SLAM2 (Monocular) [16] we track the UAV movement and use the sparse point cloud to identify the plane of the floor. For this, we apply the RANSAC algorithm on the lower half the point cloud, which is determined from UAV movement and orientation provided by ORB-SLAM2. We applied a variant of DeepLab-ResNet with 27 classes (which are condensed and more generic compared to the original 150 classes) for the segmentation of floors on images, taken by the UAV, which were also selected as keyframes by ORB-SLAM2. Utilizing the known camera poses from ORB-SLAM2 we then projected the pixels of a segment on the previously determined 3D floor pane. Figure 6 illustrates an example for the outcome of the three main steps.

While it is possible to create a dense map of the floor below a UAV trajectory with our approach, one has to make certain assumptions about the environment, which limits its usability. In the first place the floor must have enough features to be identifiable. There must also be a single, regular floor plane. Multiple floor segments on different planes would need clearly defined borders which depends highly on the quality of the results of the SLAM and semantic segmentation (this was typically not given in our scenario). Therefore, we will most likely concentrate on depth estimation based methods in the future to provide a human-readable map see VIII.

VIII. Depth Estimation

The choice of small, lightweight UAVs which only provides video input affects 3D scene modelling, SLAM or collision avoidance. In addition, local 3D scene modelling can help to improve situational awareness on USAR missions. For example, even trained experts often are in trouble with transitions of tight doorways or open windows as the estimation of the UAVs dimension in the scene is a surprisingly demanding cognitive task. This becomes even more immanent in stressful situations on USAR missions. Also, if no omnidirectional camera is used, the surrounding environment of the UAV is only partially visible. This may cause potential dangers to be occluded. In recent years DNNs to estimate depth from 2D images became increasingly popular among AI researchers [17], [18], [19]. There is an established pool of freely available datasets for training and testing, such as the NYU Depth v2 for indoor scenes [20]. However, the accuracy of these methods often depends on the scene. This is partly due to the problem statement itself, since it is ill-posed in nature, limiting the potential for 3D sensor substitution. Fuhrman et al. used a recent depth estimating DNN for collision avoidance on UAVs [21]. As stated by the authors, the DNNs instable accuracy led to collisions, even though some obstacles have successfully been avoided.

We used DenseDepth, which was amongst the most accurate methods according to the NYU Depth v2 dataset [22] at the time of this paper, and tried to push the network to be more accurate in corridor scenes specifically. We tried so by two means:

1) We created ∼ 2400 RGB-Image/Depth Image pairs from corridors with a Microsoft Kinect v1 and added them to the NYU Depth v2 training data set (∼ 10000 pairs)

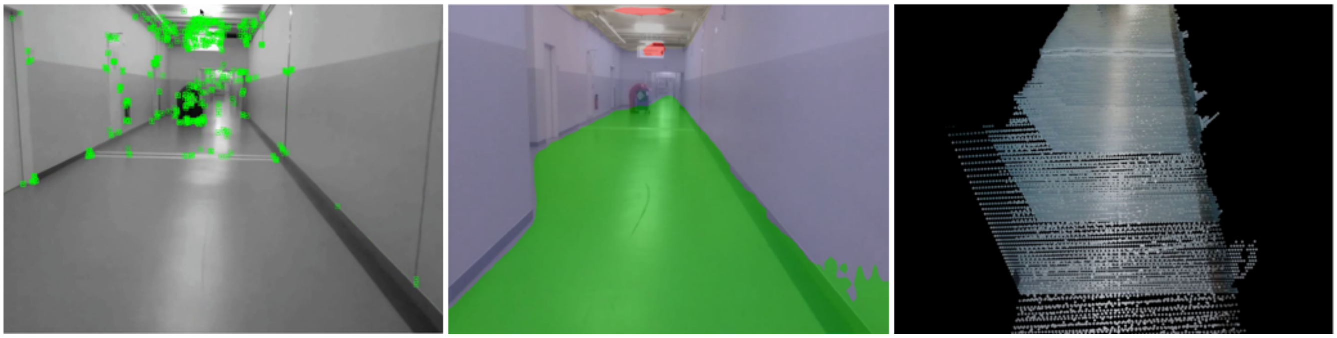

Fig. 6. Left: Feature Detection from ORB-SLAM, Center: Segmentation of the Floor with a DCNN, Right: Projection of the segment onto sparse Pointcloud

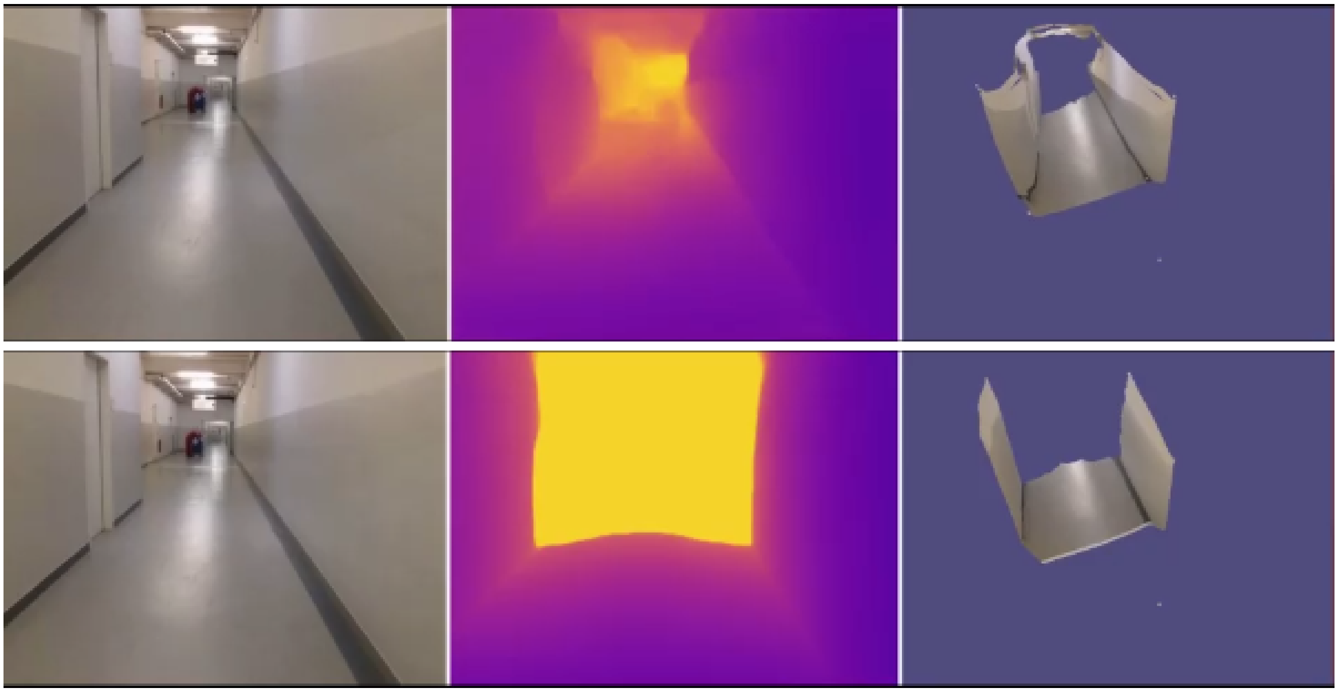

Fig. 7. Results of DenseDepth before (top row) and after refined training (bottom row). From left to right each row shows the original image input, Depth image from inference and the resulting 3D point cloud. Notice that we limit the range of the inferred depth of our optimized DenseDepth (bottom row) since the range of the Kinect v1 is limited to $\sim 4m$. Direct comparison of the Depth estimation before and after retraining shows a clear improvement in accuracy in corridor scenes.

2) We exploited the special scene structure in the training process, which is dominated by floors and walls, that is, by planes.

We created 3D point clouds from depth images and extracted the largest planar segments (in many cases, these segments were eventually walls and floors). Because of the RGB-/Depth correspondence, a mapping is given from 3D to pixels in the RGB image. This allowed us to create binary images indicating planes in the RGB image. We used these binary images as weighted masks μB_{Mask} for the depth term in the loss calculation of DenseDepth:

$$L(y,\hat{y}) = \lambda(1+\mu B_{Mask})L_{depth}(y,\hat{y}) + L_{grad}(y,\hat{y}) + L_{SSIM}(y,\hat{y})$$

We set λ to 0.1 and μ to 1, but we have not investigated other values yet. Without a claim of completeness, it just was our expectation that the adjustment of the loss term increases the pressure on planes during training. Although this is somewhat vague and doesn't hold scientific standards, we nonetheless feel that it is worth to report our current progress here. We trained an already pretrained DenseDepth for another 40 Epochs. All other training parameter remained unchanged to [18].

So far, we evaluated the results empirically by observing the output in corridor scenes, which had not been part of the training data set. In situations where the camera was moved in a straight line through corridors, we observed that the accuracy of our retrained version of DenseDepth improved over the original version (see fig. 7). We observed that the result gets unstable though in situations where the camera has an orthogonal view to walls, e.g. when the UAV moves

around a corner. In our opinion, this is mainly due to a lack of examples in the training data.
Eventually, we think that it is worth to further investigate this matter and to follow up with a clean evaluation to prove our claim. In the future we also plan to transfer our approach to other DNNs which may be either faster and more compact or more accurate or both, and to create new indoor datasets for retraining. We want to evaluate other current depth estimating DNNs in terms of their potential for real-time collision avoidance and local and global 3D scene modelling and localisation.

IX. CONCLUSIONS

Small UAVs (< 40 cm diagonal) enable USAR forces to enter buildings through small holes, open doorways and windows. Equipped with 2D cameras they deliver live video streams of scenes inaccessible for humans and series of images for later 3D scene modelling and inspection. The control of UAVs via 2D images however is cognitively demanding resulting in reduced focus on scene observation. In practice the cooperative work of two individuals is often required. Reduction of cognitive load is therefore useful. In this report we described our current progress in the development of a system which consists of one or multiple small UAVs for indoor flight, a scenario for an indoor rescue mission and a set of recent AI methods from very active fields and eventually a distributed system which distributes data across a backend of computation nodes. Our main objective was the creation of a system which allows us to quickly test new AI methods of potential interest for USAR indoor missions on live data of an UAV and under practical conditions, including the interaction of a multitude of nets. Our expectation here is to better keep track with the current state of the art. In our first scenario we have implemented various methods to provide assistence on multiple levels for the operator, ranging from guided autonomous navigation to improved spatial awareness through local 3D scene modelling. All these methods have been integrated in a distributed system architecture which features quick integration of both, methods and UAVs. Further methods can later be added or existing ones can later be exchanged if desired. The rapid integration of methods allows for quick evaluation of these methods how they interact. In the future we plan to invest more work in our current AI methods and the expansion of interfaces to integrate more sophisticated commercial lightweight UAVs.

REFERENCES

[1] https://en.wikipedia.org/wiki/August_2016_Central_Italy_earthquake.

[2] I. Kruijff-Korbayov, 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.

[3] C.-E. Hrabia, A. Hessler, Y. Xu, J. Seibert, J. Brehmer, and S. Albayrak, "Efffeu project: Towards mission-guided application of drones in safety and security environments," *Sensors*, vol. 19, no. 4, p. 973, Feb 2019. [Online]. Available: http://dx.doi.org/10.3390/s19040973

[4] A. Jacoff, "Measuring and comparing small unmanned aircraft system capabilities and remote pilot proficiency," National Institute of Standards and Technologz, Tech. Rep., 2020. [Online]. Available: https://www.nist.gov/el/intelligent-systems-division-73500/standard-test-methods-response-robots/aerial-systems

[5] 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.

[6] H. Surmann, R. Worst, T. Buschmann, A. Leinweber, A. Schmitz, G. Senkowski, and N. Goddemeier, "Integration of uavs in urban search and rescue missions," in *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Sep. 2019, pp. 203–209.

[7] H. A. Lauterbach, C. B. Koch, R. Hess, D. Eck, K. Schilling, and A. Nchter, "The eins3d project instantaneous uav-based 3d mapping for search and rescue applications," in *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2019, pp. 1–6.

[8] G. Croon and C. De Wagter, "Autonomous flight of small drones in indoor environments," 10 2018.

[9] A. Giusti, J. Guzzi, D. C. Cirean, F. He, J. P. Rodrguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. D. Caro, D. Scaramuzza, and L. M. Gambardella, "A machine learning approach to visual perception of forest trails for mobile robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 661–667, July 2016.

[10] V. Nekrasov, C. Shen, and I. D. Reid, "Light-weight refinenet for real-time semantic segmentation," *CoRR*, vol. abs/1810.03272, 2018. [Online]. Available: http://arxiv.org/abs/1810.03272

[11] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–308, September 2009, printed version publication date: June 2010. [Online]. Available: https://www.microsoft.com/en-us/research/publication/the-pascal-visual-object-classes-voc-challenge/

[12] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: object detection via region-based fully convolutional networks," *CoRR*, vol. abs/1605.06409, 2016. [Online]. Available: http://arxiv.org/abs/1605.06409

[13] "Iso 7010," https://www.iso.org/obp/ui/#iso:std:iso:7010:ed-3:v2:en.

[14] "Globally harmonized system of classification and labelling of chemicals (ghs)," http://www.unece.org/trans/danger/publi/ghs/ghs_welcome_e.html.

[15] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[16] Mur-Artal, Raúl, Montiel, J. M. M., 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, citation as specified on https://github.com/raulmur/ORB_SLAM2.

[17] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," *CoRR*, vol. abs/1411.4734, 2014. [Online]. Available: http://arxiv.org/abs/1411.4734

[18] I. Alhashim and P. Wonka, "High quality monocular depth estimation via transfer learning," *CoRR*, vol. abs/1812.11941, 2018. [Online]. Available: http://arxiv.org/abs/1812.11941

[19] J. H. Lee, M. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," *CoRR*, vol. abs/1907.10326, 2019. [Online]. Available: http://arxiv.org/abs/1907.10326

[20] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Computer Vision – ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 746–760.

[21] T. Fuhrman, D. Schneider, F. Altenberg, T. Nguyen, S. Blasen, S. Constantin, and A. Waibe, "An interactive indoor drone assistant," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019. [Online]. Available: http://dx.doi.org/10.1109/IROS40897.2019.8967587

[22] https://paperswithcode.com/sota/monocular-depth-estimation-on-nyu-depth-v2.