2010 Sixth International Conference on Autonomic and Autonomous Systems

Combining Autonomous Exploration, Goal-Oriented Coordination and Task Allocation in Multi-UAV Scenarios

Christoph Rasche, Claudius Stern, Willi Richert, Lisa Kleinjohann and Bernd Kleinjohann
C-LAB
Universität Paderborn, Fakultät für Elektrotechnik, Mathematik und Informatik
Paderborn, Germany
claudis@c-lab.de

***Abstract*—Successful rescue operations after big accidents or natural disasters require a fast and efficient overview of the overall situation. With recent advances, unmanned aerial vehicles (UAVs) are more and more a viable choice under such circumstances.**

With the number of employed UAVs, the problem of coordination arises as well as proper task allocation among possibly heterogeneous UAVs.

This paper presents a hybrid approach for UAV coordination and covers the exploration of unknown terrains as well as goal-oriented coordination and simultaneous task allocation. The approach combines the simplicity of the gradient method with informed A* search and supports prioritized task assignment. The system is suited for highly dynamic environments requiring frequent path recalculations.

***Keywords*-Path planning, Multiple UAV, Exploration, Coordination**

I. Introduction

Over the last years, unmanned aerial vehicles (UAVs) received increasing attention in rescue operations. They can be used to explore terrains and are able to achieve tasks, like firefighting in skyscrapers – dangerous and risky tasks for humans.

A major task in rescue operations is the exploration of the disaster area. This is important to get an overview of the surroundings and to locate victims. It is necessary to obtain such an overview very quickly to start effective rescue operations and – in some cases – to avoid an escalation of the situation.

Exploratory navigation includes determining all obstacles and goals in a given environment. UAVs travel from their initial positions to one or more goal positions while avoiding obstacles and recognizing landmarks and objects. Thereby, the UAVs have to memorize the explored space to plan efficient paths to unknown territory. The main challenge is to design a system which covers serious aspects, including autonomous exploration, flying to given targets, and the coordination of the multiple UAVs amongst each other. So, the system to be designed has to manage exploratory navigation as well as goal oriented planning and has to be efficient. Also a task allocation amongst the UAVs is needed.

In this paper, an approach combining exploratory navigation, goal-oriented path planning and simultaneous task allocation is presented. It is based on artificial potential fields [1] (see Section II-B) using harmonic functions [2] (see Section II-C). Path generation then utilizes an informed search algorithm in combination with a gradient method (see Section III-C). The system ensures a complete exploration of unknown regions and scales well in relation to the number of operating UAVs. By exploiting properties of linear quadtrees [3] and by using an activation window the computational costs can be reduced such that an online calculation of the whole path planning is feasible (see Section IV-C).

The presented system has two modes of operation, which can change automatically during runtime: *goal-oriented* and *non-goal-oriented* mode. Every single UAV has an individual configuration (see Section IV-B) amongst others containing its task list. In goal-oriented mode a UAV has one or more prioritized tasks (goals). In non-goal-oriented mode exploration of the whole territory will be done. The approach is cost efficient because of the combination of the advantages of goal-oriented path planning with the simplicity of gradient methods for exploration. UAVs without a given goal always move to the nearest unexplored region – in terms of path length – to optimize exploration. Local as well as global information are used to coordinate multiple UAVs and to generate consistent trajectories. Areas which probably contain important information can be explored first.

To even support the exploration in goal-oriented mode, there is an adjustable tradeoff for the maximum allowed deviation from the shortest path to explore unknown regions (see Section III-A). In many cases, gathering new information is also important and a small detour is worth a slight increase of flight time.

II. Related Work

Exploration of unknown terrain is one of the most important tasks of UAVs. Sawhney et al. [4] presented an exploration system for multiple UAVs. Their work is focused on an asynchronous exploration strategy in unknown terrains.

978-0-7695-3970-6/10 $26.00 © 2010 IEEE
DOI 10.1109/ICAS.2010.16

The main goal is to reduce the exploration time as much as possible.

Another part of UAV navigation is goal-oriented path planning. Jung et al. [5] showed a hierarchical path planning approach including a control algorithm. Nikolos et al. [6] showed a goal-oriented approach, based on evolutionary algorithms, which is capable to navigate through explored and unexplored terrain to a given goal.

All these approaches consider either goal-oriented or non-goal-oriented path planning. In this work we present a combined approach, to consider all three aspects: exploratory navigation, goal-oriented path planning and simultaneous task allocation. For example in [4] no task allocation is done and the UAVs are considered to be homogeneous (at least in terms of exploration capability). In contrast, the UAVs in this work are considered to be heterogeneous and, in particular, some tasks can only be solved by a matching UAV. Additionally, in this work exploration is needed multiple times as data becomes obsolete over time. In contrast to most path planning approaches where the optimization of the path length is focused, in this paper in some circumstances detours are allowed or even desired (see Section III-A).

A. Path Planning

In this work the presented path planning approach is considered to be a centralized one although it could also be implemented in a decentralized manner.

On the one hand UAVs have to explore unexplored parts of the terrain, while objects and landmarks must be recognized. Their position and size has to be noticed. Additionally the UAVs must keep track of regions already visited to plan efficient paths to still unexplored regions of the territory. On the other hand, UAVs have to find collision free paths from their initial positions to given goal positions. In both cases they act autonomously without any human interaction.

According to [7], it is possible to divide motion control into three general categories: global, local and reactive path planning. This work presents a combined approach of global and reactive path planning for exploratory navigation. Additionally, the global approach is used to find paths to designated goal positions.

B. Artificial potential fields for motion planning

The idea of using artificial potential fields for motion planning [1] is that paths can be obtained through linear superpositions of fictitious forces or their potentials which affect the UAV. So, a field of forces that affects the complete terrain has to be implemented. Two types of potentials exist:

1) Attractive forces, which represent goals and pull the UAV towards them.
2) Repulsive forces, which represent obstacles and push the UAV away from them.

In such a field of forces a path follows the negative gradient within the potential field. It keeps the UAV away from obstacles and pulls it towards the goal.

However, there are some problems with such potential fields, especially in case of complex and highly dynamic environments. One of the most important problems are local minima. Inside a local minimum, every potential in the neighborhood has a higher value, so the driving force vanishes and the UAV gets trapped. This can be circumvented by using artificial potential fields with harmonic functions [2].

C. Harmonic Functions

Harmonic functions satisfy Laplace's Equation in n dimensions:

$$\nabla^2\phi = \sum_{i=1}^{n} \frac{\partial^2\phi}{\partial x_i^2} = 0.$$

Namely, the second partial derivations of ϕ must be zero. The value of ϕ is given on a closed domain Ω in the configuration space $\mathcal{C}$ (see Section III) and satisfies the min-max principle and the uniqueness principle. The min-max principle guarantees the absence of local minima and the uniqueness principle the absence of flat regions.

Connoly [8] showed that it is possible to generate paths without spurious minima. Combining harmonic functions with Dirichlet's boundary conditions leads to a value-restricted configuration space, which is important for calculations with discrete arithmetic.

Originally, harmonic functions for path planning were used only with a fixed map and one single known goal. But they also suit well to dynamic environments with multiple UAVs and goals.

III. Calculation of the potential fields for path planning

We assume that the control station, responsible for the coordination of the UAVs, provides a single configuration space $\mathcal{C}$ which is used as a representation of the whole scenario. $\mathcal{C}$ is based on a quadtree (see Section IV-A), whereby the leafs carry descriptive information, like a potential value. Additionally it includes all UAVs with their properties (see Section IV-B), every goal and obstacle.

The intention is to maintain only one harmonic function which describes the whole potential field concerning multiple goals as well as multiple UAVs, avoiding the negative effects of linear superpositioning of harmonic functions described in [2]. To make use of the advantages of harmonic functions, in particular the absence of local minima, the potential values of the configuration space have to fulfill several requirements, e. g., potential values must strictly increase with the distance to the goals. To take into account all these restrictions, the calculation process is modeled as an optimization problem. So, a first approximation has to be computed and afterwards an iterative relaxation technique is used for successive enhancement.

A. First Approximation

For a first approximation the values for targets are set to zero and the values for obstacles are set to one. To calculate the remaining values the following equation is used.

$$\phi(x,y) = \xi \cdot \frac{\log\left(\tau\left(x,y\right)\right)}{\log(d)}$$

$\phi(x,y)$ satisfies Laplace's equation. Hereby, $\tau(x,y)$ represents the euclidean distance from the point $(x,y) \in \mathcal{C}$ to the nearest target point. The logarithm of the diagonal d of the whole terrain is used to normalize the values between 0 and 1. For goal-oriented mode, it is possible to trade off between shortest path calculation and additional information the UAV could gather while taking a detour over unexplored regions. For this the ξ-Value was introduced. The value is between 0 and 1 and is set by the user. The less ξ is, the more attractive unexplored terrain is to the path planner.

As mentioned, the equation is used to calculate a first approximation for every unbound potential in the environment. This leads to a linear system of equations.

B. Update equation

The first approximation will be successively enhanced using a relaxation method. In this work the Gauss-Seidel algorithm was used. While using a quadtree to partition $\mathcal{C}$, the dimensions of the part of the terrain each leaf represents have to be considered. This leads to the following update equation, which is based on [9]:

$$\begin{aligned}\phi(x,y) = \quad & \frac{\tau_{up}\tau_{down}[\tau_{right}\phi(x+1,y)+\tau_{left}\phi(x-1,y)]}{\tau_{up}\tau_{down}(\tau_{right}+\tau_{left})+\tau_{right}\tau_{left}(\tau_{down}+\tau_{up})} \\ + \quad & \frac{\tau_{right}\tau_{left}[\tau_{down}\phi(x,y-1)+\tau_{up}\phi(x,y+1)]}{\tau_{down}\tau_{up}(\tau_{right}+\tau_{left})+\tau_{right}\tau_{left}(\tau_{down}+\tau_{up})}.\end{aligned}$$

Here, $\tau_{up}, \tau_{down}, \tau_{left}, \tau_{right}$, represent the distances from the center of the current leaf (x,y) to the center of the according neighbor. Neighbor leafs are all leafs which have a border in common. Hence, also possibly different leaf sizes are taken into account. The potential value of the current leaf is represented through $\phi(x,y)$, the potential of the left neighbor through $\phi(x-1,y)$ and so on.

With a given error rate $\epsilon \leq 10^{-p}$ and M leafs, $\mathcal{O}(pM)$ iterations are needed. Intuitively, choosing a quadtree instead of a grid lowers the calculation costs. However, for large terrains, together with the demand for detailed resolution, this method is still very costly.

Amongst others, several methods reducing these costs are presented in the following section.

C. Path Planning Approach

Based on the potential field stored in the configuration space $\mathcal{C}$ as described in the previous section, discrete path planning is possible. Here two different approaches are utilized: gradient based path planning for exploratory navigation (non-goal-oriented) and the A*-algorithm for goal-oriented path planning. The used UAVs are heterogeneous, e. g., in terms of equipment. So, each UAV has to be able to reach every point of the terrain. Therefore it is impossible to split up the terrain into subareas, assigned to a single UAV.

The information about the used terrain can be obtained in two ways. One way is to get the data through third party applications like satellite images. Additionally the UAVs have on-board cameras for the exploration of unknown areas and to update the information available so far.

When no explicit goals, which have to be achieved, are given, the negative gradient $-\nabla\phi$ of the potential field is used for exploratory path planning. Otherwise A* is utilized for goal-oriented path planning.

If only a gradient method would be used for goal-oriented path planning, the configuration space had to be recalculated for each UAV and each path to an explicit goal. In scenarios consisting of highly complex environments with multiple UAVs this is very costly. If a UAV detects that the current path has been interrupted, it automatically generates a new one, implying a recalculation of $\mathcal{C}$.

IV. Construction of the Configuration Space

In order to represent the complete scenario, the configuration space has to contain the coordinates and identities of all the drones, all the goals, the obstacles and unknown areas. This section, therefore, describes how the complete area is discretized into cells. Afterwards, it is presented how additional information regarding the drones, like fuel level or individual tasks, is represented. Finally, the section discusses how the path recalculation complexity can be decreased to a level, at which online calculation is feasible.

A. Cell decomposition

As mentioned before, the configuration space is divided by a quadtree and each UAV is represented as a point in $\mathcal{C}$. First $\mathcal{C}$ is divided into free space $\mathcal{C}_{free}$ and obstacle space $\mathcal{CB}_i$. After that the obstacle space is expanded by the dimensions of the UAVs. Finally, the resulting $\mathcal{C}_{free}$ is subdivided into unknown, unexplored, explored and goal space. This space types are represented through different leaf types. This subdivision changes dynamically during runtime. A distinction between unknown and unexplored space was made. Unknown space represents areas with no information about the terrain height. Unexplored space represents areas without up-to-date information. Hence, during a rescue mission the age of some data is of crucial importance. If the exploration time of a region exceeds a pre-defined threshold, the according region becomes unexplored again but it can never become unknown again.

The quadtree divides the whole space into four subspaces of equal size. These subspaces are recursively divided into four subspaces. Subdivision stops when the enclosed space of a leaf is of homogeneous type or a pre-defined maximum breakdown is reached. Homogeneous means that the space consists only of one type, e. g. unexplored space. If the given breakdown is reached and the space is not homogeneous, an approximation will be done: Every leaf which includes occupied space will be marked as occupied space. Otherwise, if the space includes different types of $\mathcal{C}_{free}$, it will be marked with that type of space which constitutes the largest part of the leaf. After subdivision, the complete terrain is represented through the leafs of the quadtree.

To specify the positions of UAVs and obstacles, a Cartesian coordinate system was embedded into the configuration space $\mathcal{C}$. Hence, each node of the quadtree has an associated position and dimensions within the coordinate system.

B. UAV configuration

Each UAV has a configuration within $\mathcal{C}$, which consists of a position related to the coordinate system, its direction, a task list, its role, and finally its current fuel level.

The position of the UAVs in $\mathcal{C}$ is described by three entries and must be unique, because if two UAVs had the same position in $\mathcal{C}$ they would have crashed. For path planning only two entries are used. The height of the UAVs is neglected as they have a favorite height over ground. Of course the height is important to avoid collisions. It is also important for the cameras of the UAVs. The area the UAVs can explore at one moment depends on the flare angle of their cameras and the height of the UAVs. Additionally every UAV has a maximum altitude. This altitude is used to partition the terrain into freespace and obstacle space. Each part of the terrain which is higher than the maximum altitude is obstacle space, the remainder is freespace.

The configuration of a UAV contains a flight direction. This direction can also be used to provide a preferred direction in which the UAV will fly first while exploring. The modeled UAVs can turn on the spot so the direction is not too important for motion planning. Previous tests showed that in this case the use of a preferred direction does not lead to better results, concerning the exploration rate or speed. Because of that there is no favorite direction for the UAVs in the resulting system. The direction in combination with the position is used to describe the single UAVs.

A task always contains one known goal. It can be UAV-specific or non-UAV-specific, e. g., refueling would be a UAV-specific task. Goals can be points which the UAV has to explore or areas which have to be monitored with high priority. Every goal is associated with one task. When the fuel level reaches a given threshold, the UAV sets up a new task with a fuel station as goal and highest priority. This task displaces all other tasks to avoid damage.

Every UAV also has a role, which is used to realize a scheduling of all given tasks. In this work three roles have been defined: explorer, seeker, surveillant. A role is associated with a specific priority scheme. Depending on the role the task list will be sorted according to the different task priorities. Explorer UAVs first explore the terrain concerning specific goals with lower priority. Seeker UAVs favor goal points over areas and exploration. Surveillant UAVs favor goal areas over points and exploration. The priority schemes of the seeker and the surveillant roles initialize tasks like goal points or goal areas with high priority to favor these tasks over exploration. To avoid starvation, the priority of low priority tasks increases after given time intervals.

C. Cost reduction

The methods presented in the previous sections ensure complete exploration, reaching of given goals and task allocation. However, these methods are rather complex, especially the recalculation of $\mathcal{C}$. Hence, without any cost reduction techniques they do not suit well for embedded systems.

Recalculation of $\mathcal{C}$ will be done for the following reasons:

- Insertion of new targets and obstacles
- Deletion of targets and obstacles
- Path-planning request after exploration

After a UAV changed the configuration space through exploration it requests a new path. In that case the central control station recalculates the potential values of the global configuration space. Obviously, this leads to frequent recalculations of $\mathcal{C}$. In addition, the server load depends directly on the number of active UAVs. Because of these reasons, the recalculation step must be performed efficiently.

The costs to calculate the configuration space are directly related to the number of nodes to be updated. Hence, a quadtree was used instead of a grid, which is used in the majority of cases to represent a terrain. To reduce the costs for neighbor finding in the tree – which is necessary for the update equation and the path planning methods – the tree was extended to a linear quadtree. To transform a tree into a linear quadtree a unique code and its depth is saved for each leaf. An algorithm which finds neighbors in mean time of $\mathcal{O}(1)$ was implemented. This is a further improved version of [3] with the focus on reducing tree editing costs. With only one configuration space for all UAVs, it is crucial to have a data structure which can be modified easily – usually this is done several times per second. In contrast to [3] no additional information about the depth difference to the neighbors is stored in the nodes. This significantly increases the editing speed.

To reduce the calculation costs for the configuration space even more, break conditions were established. The iterative calculation stops, if:

- the potential values remain unchanged
- the number of local minima is less than 0.5% of the number of updated leafs
- the iteration depth is greater than or equal to 10% of the number of updated leafs

The last condition guarantees the termination of the update method. Tests showed that these break conditions lead to an appropriate tradeoff between number of local minima and calculation costs.

Furthermore, leafs with potential values greater than or equal to 0.999 are not considered as local minima. Tests showed that such values occur most times in the neighborhood of newly explored obstacles. Usually such potential values are greater than the potential value of areas which contain the UAVs. This leads to a significant reduction of the calculation costs. If a UAV gets trapped in a local minimum the A*-algorithm is used to leave it.

For further cost reduction in non-goal-oriented mode, an activation window was established. First all nodes are inactive. After the exploration of an area the according leaf is marked as active. After a given time leafs can be marked as unexplored again and are not active any longer. The first approximation will be calculated only for active leafs.

V. Results

The following tests were done to demonstrate that the system assure a complete exploration of a given terrain. Even by using only a single configuration space the use of multiple UAVs leads to faster exploration rates. Additionally, by using the cost reduction methods introduced in the previous section all calculations can be done online. For testing a 3D simulation was created which represents the terrain with the corresponding configuration space and the UAVs. The simulations were run on a desktop PC with a dual core 2.6 GHz CPU.

The tests were executed on a fictive map with simulated UAVs. The terrain was represented through a height map with a resolution of 256 × 256 pixels. The underlying quadtree had a maximum resolution of 2×2 pixels. An area of $1000m \times 1000m \times 160m$ was simulated. The UAVs had a maximum altitude of $140m$. So, everything above $140m$ was treated as occupied space. In our test scenarios $\mathcal{C}$ was partitioned into 72% $\mathcal{C}_{free}$ and 28% $\mathcal{CB}_i$. The UAVs flew with a speed of 60km/h and had a favorite height of 40m over ground. To explore the terrain the UAVs used a camera with a flare angle of $90°$. By using such a camera the UAVs were able to explore $5026, 55m^2$ at one moment when they had reached their favorite height.

Four tests are presented in this paper. They consist of exploring the terrain. The exploration was done by 1, 3, 6 and 9 UAVs. The partitioning of the terrain was unknown at the beginning. The UAVs had to recognize the occupied space through their cameras. For all tests the exploration was non-goal-oriented. Therefore, no areas had to be explored with high priority and the exploration was continued as long as unexplored terrain existed. Mostly, the gradient method was used to reduce the costs for path planning. Additionally, the A*-algorithm was used whenever UAVs got trapped in local minima.

Figure 1 shows the percentage explored terrain covered, related to the exploration duration. The duration is given in real time. In every test the complete terrain was explored. By using three UAVs the exploration time was reduced to the half instead of using one UAV. Increasing the number of UAVs always leads to a lower time needed for exploration. But tripling the number of UAVs to nine did not reduce the exploration time to the half again, because the more UAVs were used the more often they constrained each other and collisions had to be avoided. This needs time in which the UAVs do not explore much more terrain. Another point is that the less unexplored terrain is left the more UAVs start to fly to the same terrain to explore it.

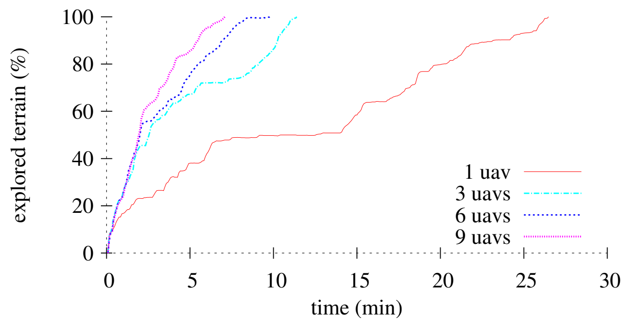

Figure 1. Percentage terrain covered

Additionally, the figure shows that until an exploration rate of 70% was reached there are only a few time spans in which the UAVs did not explore large areas of the terrain. This lack-of-exploration behavior occurs after the UAVs explored a large terrain and had missed a few small areas or when the next unexplored area was far away from the UAVs.

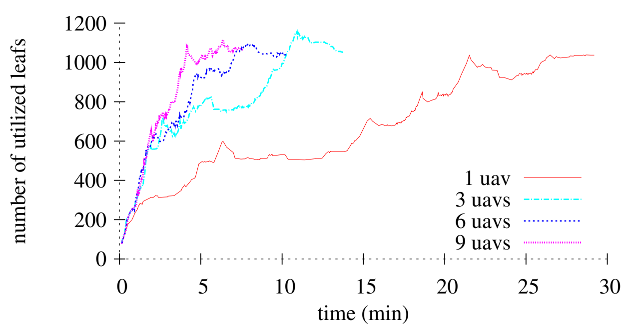

Figure 2. Number of utilized leafs for the calculations of $\mathcal{C}$

Figure 2 shows the number of explored leafs related to the exploration duration. These leafs are those which have to be updated. It shows that a maximum of 1200 leafs had to be updated. The number of used leafs increased very fast at the beginning and decreased after a while because of recombining the quadtree. As shown in the figure the number of leafs is independent from the number of used UAVs.

Figure 3 shows the costs for calculating the complete configuration space once. The calculations contain the following tasks:

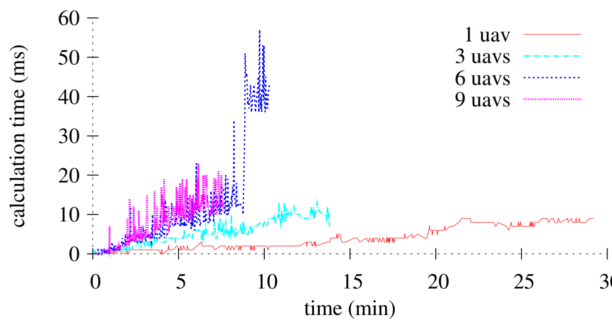

Figure 3. Calculation times for single calculations of $\mathcal{C}$

- Assignment of the next terrains to be explored
- Bounding of target and occupied space
- Calculation of the distances to the next target space
- Calculation of the first approximation
- Determining all neighbors of the relevant leafs
- Update of the potential values
- Check for break conditions

The costs raise with the exploration rate because of the activation window. The more areas are explored, the more leafs are active and have to be updated. But besides this the costs are relatively constant except of a few runaway values. They increase with the number of UAVs. This is because the more UAVs are used the more complex the environment becomes since more UAVs have to be considered. Additionally more UAVs lead to more recalculations, which also increases the calculation costs. But the costs are in most cases below 20ms which shall allows online calculations.

VI. Conclusion and Future Work

The motivation for this paper was to design a path planning system within the research project SOGRO[1]. The requirement was to create an efficient and robust system for coordination of multiple UAVs, including path planning, exploratory navigation, and simultaneous task allocation, using only one global configuration space.

A hybrid approach for UAV coordination and efficient exploration of disaster areas was presented. It uses artificial potential fields, combined with an informed search algorithm, and a role system. Additional methods like a quadtree, an activation window and break conditions were used to find a tradeoff between the number of local minima and computational costs. The results show that until an exploration rate of more than 70% was reached, a nearly constant exploration rate was achieved. Three UAVs need only half of the time for exploration in comparison to the time one single UAV needs.

The costs to compute the configuration space were decreased such that an online calculation without any noticeable delay was possible. This is important for highly dynamic environments, where the calculation has to be faster than the changes of the configuration space, in order to calculate efficient paths. In combination with the total exploration of the terrain, this leads to a robust and efficient system.

Future work is to lower the exploration time even more when using multiple UAVs. This can be done by more active coordination methods, e. g., explicit communication between the UAVs. Another focus of future work is to achieve greater autonomy. For this purpose an interaction of the UAVs with each other will be implemented to replace the central control station. This should be done in such a way, that using more UAVs leads to even faster exploration rates.

The next step of the research project is to apply the approach to real UAVs. The achieved cost reduction enables the system to be implemented even on embedded systems. Of course, physical properties of real UAVs have to be considered even more. Additionally, sensor errors and the UAV behavior, e. g., in strong crosswind may lead to further need for adaptations when using real UAVs.

[1]"Sofortrettung bei Großunfall mit Massenanfall von Verletzten", supported by the Bundesministerium für Bildung und Forschung

References

[1] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *IJRR*, vol. 5, no. 1, pp. 90–98, 1986.

[2] C. Connolly, J. Burns, and R. Weiss, "Path planning using laplace's equation," *IEEE ICRA*, vol. 3, pp. 2102–2106, Mai 1990.

[3] K. Aizawa, K. Motomura, S. Kimura, R. Kadowaki, and J. Fan, "Constant time neighbor finding in quadtrees: An experimental result," in *ISCCSP*, March 2008, pp. 505–510.

[4] R. Sawhney, K. M. Krishna, and K. Srinathan, "On fast exploration in 2d and 3d terrains with multiple robots," in *AAMAS*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 73–80.

[5] D. Jung, J. Ratti, and P. Tsiotras, "Real-time implementation and validation of a new hierarchical path planning scheme of UAVs via hardware-in-the-loop simulation," *Journal of Intelligent and Robotic Systems*, vol. 54, no. 1-3, pp. 163–181, March 2009.

[6] I. Nikolos, K. Valavanis, N. Tsourveloudis, and A. Kostaras, "Evolutionary algorithm based offline/online path planner for uav navigation," *IEEE SMC*, vol. 33, no. 6, pp. 898–912, Dec. 2003.

[7] D. Beck, A. Ferrein, and G. Lakemeyer, "Landmark-based representations for navigating holonomic soccer robots," in *RoboCup 2008: Robot Soccer World Cup XII*, ser. Lecture Notes in Computer Science, vol. 5399. Springer Berlin / Heidelberg, 2009, pp. 25–36.

[8] C. Connolly, "Applications of harmonic functions to robotics," in *IEEE ISIC*, Aug 1992, pp. 498–502.

[9] J. S. Zelek, "A framework for mobile robot concurrent path planning and execution in incomplete and uncertain environments," in *AIPS*, 1998.