Surveillance Planning with Localization Uncertainty for UAVs

Jan Faigl*, Tomáš Krajník, Vojtěch Vonásek, Libor Přeučil
Department of Cybernetics, Faculty of Electrical Engineering
Czech Technical University in Prague
{xfaigl,tkrajnik,vonasek,preucil}@labe.felk.cvut.cz, *corresponding author

Abstract—

This paper presents a new multi-goal path planning method that incorporates the localization uncertainty in a visual inspection surveillance task. It is shown that the reliability of the executed found plan is increased if the localization uncertainty of the used navigation method is taken into account during the path planning. The navigation method follows the map&replay technique based on a combination of monocular vision and dead-reckoning. The mathematical description of the navigation method allows efficient computation of the evolution of the robot position uncertainty that is used in the proposed path planning algorithm. The algorithm minimizes the length of the inspection path while the robot position error at the goals is decreased. The effect of the decreased localization uncertainty is examined in several scenarios.

The presented experimental results indicate that probability of the goals visits can be increased by the proposed algorithm. Thus, the proposed approach opens further research directions in the increasing reliability of the autonomous navigation by the path planning using efficient and sufficiently informative heuristics of the localization error evolution.

I. Introduction

The problem of autonomous navigation of a mobile robot is addressed by various localization methods in order to achieve sufficiently reliable navigation. The methods use various techniques based on different assumptions and environment constraints. One of the popular technique is the so-called SLAM in which no prior information about the environment is known and the localization is performed on the basis of the simultaneously created map. Even though a significant progress has been made in this field, more reliable localization techniques use a priori known map of the robot surrounding environment. Moreover, the authors of SLAM algorithms tends to consider only the immediate localization error, and it is not exceptional that robots are navigated manually during SLAM examination. On the other side of the navigation methods, higher precision and reliability (in a long term) is achieved at the cost of the previously created map. In this sense, reliable navigation methods are based on the visual servoing techniques using the map&replay scenario [1], [2].

Beside improvements of the localization methods, the reliability of the autonomous navigation may be increased by consideration of environment properties and the localization/navigation method in the preparation of the plan/path for the navigation. The idea is simple: the robot identifies areas, where the localization method would be too imprecise and avoids these places. Although the idea is simple, the problem is not easily tractable as it depends on appropriate (realistic) environment models. Moreover the problem domain has to be extended by the "uncertainty" dimensions (e.g. pose$\times$uncertainties [3]) that increase the computational complexity of the planning methods. Models of the uncertainty evolution have to be sufficiently informative otherwise the intended plan will be more likely useless. For example a model of the increasing odometry error simply leads to minimization of the planned path length, but it does not provide a way to "correct" the robot pose estimation. Therefore such model cannot be efficiently used in a long-term planning. The models based on performing a simulation of the localization within the map of the robot surrounding environment are too computationally intensive and therefore approximations have to be used. However, such simplified models may lead to violate the required stability assumptions of the localization methods [4], [5], [6].

In this paper, we consider a heuristic function describing the evolution of the localization uncertainty in a surveillance path planning problem that deals with visiting a set of areas of interest (AoIs). The function is derived from the model of a simple navigation principle [7] that uses the map&replay technique. The principle is based on a detection of salient objects and dead-reckoning measurements, e.g. an odometry, and it is similar to [8], but its localization error is bound and theoretically proven. The heuristic function is used in the modified competitive rule of the self-organizing map (SOM) approach for the Traveling Salesman Problem (TSP) [9] that is applied to the multi-goal path planning problem, i.e. the problem of finding a shortest path visiting given set of goals [10]. The proposed method finds a path with lower localization uncertainty at the visited AoIs than in the case of methods minimizing only the path length.

This paper is organized as follows. The localization uncertainty model and the derived heuristic function are described in Section II together with an overview of the used SOM scheme. The problem formulation is presented in Section III The proposed planning method is presented in Section IV. Experimental results of the proposed method are presented in Section V. A discussion of the proposed approach and remarks about future work are presented in the conclusion.

II. Related Work

A. Navigation Method and Uncertainty Model

The used navigation method [7] is based on heading corrections only while the traveled distance is estimated by a rel-

atively imprecise odometry. The heading corrections utilize salient objects (Speeded Up Robust Features (SURF) [11]) recognized in the environment in the map&replay approach. At first, the robot is tele-operated in the environment along straight line segments in the mapping phase. For each segment a set of visual landmarks is remembered and the local length of the segment is measured by the odometry, thus the long term instability of the odometry is not an issue. In the replay phase, the robot is placed at the starting segment and requested to travel the learned path. The current visible landmarks are matched with the learned ones by the histogram voting method, which realizes the so-called visual compass. Based on the heading deviation the control law steers the robot in the desired direction. Once the traveled distance reaches the segment length, the robot is turned into the direction of the next segment by the compass, and the next segment is traversed in the same manner. Even though the navigation method is very simple, it allows the robot to continuously travel closed paths more than one kilometer long in real outdoor environments in variable lighting conditions and seasonal changes, see experimental results in [7].

To establish the localization uncertainty model a simple case of a robot navigating along a single segment aligned with the x axis can be considered, see Fig. 1. Let the robot

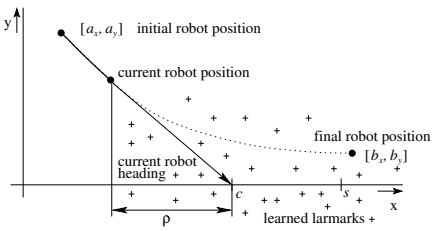


Fig. 1: Robot navigation model

start at position $[a_x, a_y]$ and learned landmarks are in front of the robot. During the robot movement, the nearby landmarks disappear and more distant landmarks become visible, and therefore landmarks can be considered in a constant distance ρ ahead of the robot. The robot movement can be characterized by the differential equation $dx/dy = \rho/-y$. Assuming $c \approx a_y$ and boundary conditions the final robot position is $[b_x, b_y] = [a_x + s, a_y \exp(-s/\rho)]$. The model of the robot movement can be augmented by an odometry error ν and heading sensor noise ξ (random variables drawn from the Gaussian distribution with the zero mean and the variances η and τ respectively) resulting in

$$\begin{bmatrix} b_x \\ b_y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{-\frac{s}{\rho}} \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix} + \begin{bmatrix} s + s\nu \\ \xi \end{bmatrix}, \quad (1)$$

which rewritten to a matrix form denotes $\boldsymbol{b} = \boldsymbol{M}\boldsymbol{a} + \boldsymbol{s}$. For an arbitrary orientation of the segment the matrix equation can be complemented by the rotation matrix $\boldsymbol{R}$:

$$\boldsymbol{b} = \boldsymbol{R}^T \boldsymbol{M} \boldsymbol{R} \boldsymbol{a} + \boldsymbol{R}^T \boldsymbol{s}. \quad (2)$$

The evolution of the robot position uncertainty for a single segment is based on consideration of the robot position as a random variable drawn from 2D normal distribution with the mean $\hat{\boldsymbol{a}}$ and the covariance matrix $\boldsymbol{A}$. Due to linear and absolute terms of Equation (2) the uncertainty at the segment end is a normal distribution with the mean $\hat{\boldsymbol{b}}$ and the covariance matrix $\boldsymbol{B}$. To investigate the evolution of the robot position covariance matrix, the robot position can be denoted to $\boldsymbol{a} = \hat{\boldsymbol{a}} + \tilde{\boldsymbol{a}}$, where $\hat{\boldsymbol{a}}$ is the mean of $\boldsymbol{a}$ and $\tilde{\boldsymbol{a}}$ denotes a random variable with the zero mean and the covariance matrix $\boldsymbol{A}$. Regarding to independence of $\tilde{\boldsymbol{s}}$ and $\tilde{\boldsymbol{a}}$

$$\tilde{\boldsymbol{b}}\tilde{\boldsymbol{b}}^T = \boldsymbol{R}^T \boldsymbol{M} \boldsymbol{R} \tilde{\boldsymbol{a}} \tilde{\boldsymbol{a}}^T \boldsymbol{R}^T \boldsymbol{M}^T \boldsymbol{R} + \boldsymbol{R}^T \tilde{\boldsymbol{s}} \tilde{\boldsymbol{s}}^T \boldsymbol{R}. \quad (3)$$

For a sequence of segments, the end of the segment i is the start of the segment $i+1$, i.e. $\boldsymbol{a}_{i+1} = \boldsymbol{b}_i = \boldsymbol{R}_i^T \boldsymbol{M}_i \boldsymbol{R}_i \boldsymbol{a}_i + \boldsymbol{R}_i^T \boldsymbol{s}_i$. The evolution of the uncertainty is a recurrent form of Equation (3) that is in terms of covariance matrices

$$\boldsymbol{A}_{i+1} = \boldsymbol{R}_i^T \boldsymbol{M}_i \boldsymbol{R}_i \boldsymbol{A}_i \boldsymbol{R}_i^T \boldsymbol{M}_i^T \boldsymbol{R}_i + \boldsymbol{R}_i^T \boldsymbol{S}_i \boldsymbol{R}_i, \quad (4)$$

where

$$\boldsymbol{M}_i = \begin{bmatrix} 1 & 0 \\ 0 & e^{-\frac{s_i}{\rho}} \end{bmatrix}, \boldsymbol{S}_i = \begin{bmatrix} s_i^2 \eta^2 & 0 \\ 0 & \tau^2 \end{bmatrix}$$

The length of the traveled segment is s_i, ρ represents density of the landmarks in the environment (an "average" distance of the landmarks to the robot), η and τ represent precision of the odometry and the heading sensor.

Equation (4) provides estimation of the robot position uncertainty at the particular segment end and can be used as a heuristic function of the uncertainty evolution in a path planning algorithm. The stability (boundness) of the navigation method for closed paths consisting of several conjoined straight line segments is proven in [7] for a robot moving in a plane with an imprecise measurements of the traveled distance, a forward aimed camera capable recognizing a nonempty subset of mapped landmarks and paths with at least two noncollinear segments.

B. Self-Organizing Map Based Multi-Goal Path Planning

The multi-goal path planning problem is formulated as the TSP, which allows to use any TSP solver. However, self-organizing map (SOM) approach is useful, because it is flexible enough to consider the evolution of the localization uncertainty in a straightforward way. The SOM algorithm for the TSP [9] has been selected as the main adaptation schema being modified. The algorithm is Kohonen's unsupervised neural network in which nodes are organized into a cycle and a solution of the TSP is represented by synaptic weights of nodes that are adapted to the goals (cities) during the self-adaptation process.

The adaptation consists of two phases: *competitive* and *cooperative*. In the *competitive* phase, goals are presented to the network in a random order. For each goal a winner (the closest) node is found according to its Euclidean distance to the goal. The *cooperative* phase is an adaptation of nodes (weights) to the presented goal. A winner node and its neighbouring nodes are adapted (moved) towards the presenting goal g by the adaptation rule $\nu_j' = \nu_j + \mu f(\sigma, l)(g - \nu_j)$, where ν_j and g are coordinates of the node and the presented

goal, μ is the fractional learning rate and $f(\sigma, l)$ is the neighbouring function. The function is $f(\sigma, l) = exp(-l^2/\sigma^2)$ for $l < d$ and $f(\sigma, l) = 0$ otherwise, where σ is the gain parameter (also called the neighbourhood function variance), l is the distance in a number of nodes measured along the ring, d is the size of the winner node neighbourhood that is set to $d = 0.2m$, where m is the number of nodes. After the complete presentation of all goals (one adaptation step) the gain is decreased by $\sigma = (1 - \alpha)\sigma$, where α is the gain decreasing rate. The adaptation process is repeated until the distance of a winner to the city is lower than given threshold, e.g. 0.001. An inhibition mechanism [9] is used to avoid nodes to win too often. The initial value of σ is set according to $\sigma_0 = 0.06 + 12.41n$, where n is the number of goals. The learning and decreasing rates are $\mu = 0.6$, $\alpha = 0.1$. The number of nodes m is set to $m = 2.5n$.

III. Problem Statement

The problem addressed in this paper is motivated by an instance of the inspection task that is a problem of visiting given set of goals. The goals' positions in the environment are known in advance. A mobile robot is requested to repeatably visit the goals, due to its restricted sensing capabilities, i.e. all goals are not visible from a single place. The problem is to find a sequence of goals visits with a minimal inspection period. The robot is capable to be navigated by the method described in Section II-A. The robot uses imperfect sensors, therefore its position estimation is imprecise.

Even though the stability of the used navigation method has been theoretically proven and experimentally verified in real-world environments, it does not mean that the localization error is sufficiently low. Therefore, to increase the probability of visiting the requested goals, the localization uncertainty at the goal positions should be as small as possible.

For simplicity, the environment is assumed to be obstacle free, thus a path can be composed from straight line segments connecting the goals. Once a path is found, the robot is navigated along the path in the tele-operated manner in order to create a map of the environment. After that, the robot is requested to periodically visit the goals using the mapped path.

IV. Planning with Localization Uncertainty

The proposed multi-goal path planning algorithm combines Equation (4) and the SOM adaptation schema for the TSP. The idea is based on evolution of the localization uncertainty that depends on particular values of η, τ and ρ. The imprecise odometry increases the error in the direction of the robot movement, while the error is suppressed by heading corrections in the lateral direction. For a robot with single forward looking camera, it means that a lower localization uncertainty is achieved by a moving along "zig-zag" trajectories instead of a single straight line segment. So, the direction from which the robot arrives to the goal is crucial in the uncertainty decreasing process. In other words, a simple straight line segment path between two goals g_1 and g_2 increases the error in the segment (longitudinal) direction due to the imprecise odometry. To decrease the uncertainty at g_2 caused by the odometry, the robot can be navigated to an auxiliary navigation point close to g_2. From a point at some perimeter around g_2, the robot movement to g_2 diminishes the previously increased error in the g_1–g_2 longitudinal direction. The situation is demonstrated in Fig. 4. A radius of the perimeter can be selected according to the odometry error and profuseness of landmarks in the environment that is represented by ρ.

Based on the observation the planning problem is formulated as the following modification of the TSP. Each goal g is represented by a group of points $P_g = \{p_{g,1}, \ldots, p_{g,k}\}$ at the perimeter d_p and the problem is to select a single point from each group such that the uncertainty at each goal is minimized and the total route length is minimized as well. The SOM competitive and cooperative adaptation rules are modified to consider the evolution of the localization uncertainty using Equation (4). The ring of nodes must be oriented in order to used the equation during the adaptation process. It is achieved by the adaptation of the first and the last ring nodes (in fact any two neighbouring nodes can be used) to the selected starting goal prior presentation of other goals to the network. During the ring evolution, each node has associated the localization uncertainty represented by the covariance matrix A_ν. The covariance matrix at the first node is computed from the connection of the node with the starting goal by a straight line segment. The initial uncertainty at the starting goal is set to zero. The matrix A_{ν_i} is computed from $A_{\nu_{i-1}}$ directly by Equation (4) where s is the Euclidean distance between the nodes ν_{i-1}, ν_i. The modified rules are as follows.

The winner node to a goal g is selected from not inhibited nodes according to the Euclidean distance between a node and the goal. The orientation of the ring defines forward and backward neighbourhoods of the winner node, which are utilized in the cooperative phase. The backward neighbouring nodes of the winner node $\nu^\star$ are adapted to the perimeter point $p_{g,\star}$, while $\nu^\star$ and its forward neighbouring nodes are adapted towards g. The perimeter point $p_{g,\star}$ is selected from P_g according to

$$p_{g,\star} = \mathrm{argmin}_{p \in P_g}(||\boldsymbol{A}_g||^2), \tag{5}$$

where $||\boldsymbol{A}_g||^2$ denotes the norm of the covariance matrix, i.e. the maximal eigenvalue of $\boldsymbol{A}_g \boldsymbol{A}_g^T$. The particular matrix $\boldsymbol{A}_g$ is computed from the straight line segments $(\nu^{\star -1}, p_{g,.})$ and $(p_{g,.}, g)$, where $\nu^{\star -1}$ is the first backward node of the winner. After the adaptation, the nodes $\nu^\star$ and $\nu^{\star -1}$ are marked as inhibited for the rest of the current adaptation step. The proposed algorithm is called *dadapt* from the "*double adaptation*", because of two performed adaptations: towards $p_{g,\star}$ and g.

The final path can be constructed by two methods. The first variant is called *dadapt-ring*, because it directly uses the ring as the final path. The second variant consider only the winner nodes associated to the goals and their first backward

nodes associated to the perimeter points, the variant is called *dadapt-perim*.

V. Experiments

The proposed multi-goal path planning algorithm has been experimentally verified in several scenarios. Four path construction variants are considered in the algorithm examination. The first variant is a solution of the TSP without consideration of the localization uncertainty and it is refereed as *simple*. The second variant represents straightforward decreasing of the longitudinal error and it is used as a reference method to examine quality of SOM solutions. The path is constructed from a TSP solution where an additional perimeter point p is placed before each visited goal. The point is placed at the perimeter in such a way, that line segments $(g_{i-1}, p$ and (p, g_i) form the right angle and one of the two solutions is randomly selected. Finally two SOM algorithm variants *dadapt-perim* and *dadapt-ring* are used.

The quality of solution is characterized by the length of the found path L and the maximal localization uncertainty at the visited goals E_g computed from the covariance matrix

$$E_{max} = \max_{g \in \boldsymbol{G}} \sqrt{(||\boldsymbol{A}_g||^2)}.$$

Due to randomization of the SOM algorithm fifty solutions are found by the particular algorithm variant and the quality metrics are computed as average values. Besides, the best solution with the lowest E_{best} is used to estimate algorithm capability to find good solutions.

Fig. 2: AR Drone quadcopter.

The considered robot is the AR Drone quadcopter [12] shown in Fig. 2. The following values for the outdoor environment have been set: η=0.1 m, τ=0.1 m and ρ=20 m. In the case of an indoor environment, landmarks are closer, therefore $\rho = 5.5$ has been used. The used SOM parameters are $\sigma_0 = 12.14n + 0.6, \mu = 0.6, \alpha = 0.1, \delta = 0.001, d = 0.2m$, where n is the number of goals and m is the number of nodes that is set to $m = 4n$. To examine the effect of the perimeter, solutions have been found for several perimeter radiuses. Each goal perimeter is sampled by 48 equidistantly placed points in the proposed SOM algorithm with dual adaptation.

A. Planned paths for Outdoor Environments

Experimental results for the *square* scenario are depicted in Fig. 3 and examples of found paths for perimeter radius 19 meters are shown in Fig. 4. The results indicate that considering localization uncertainty leads to more than two

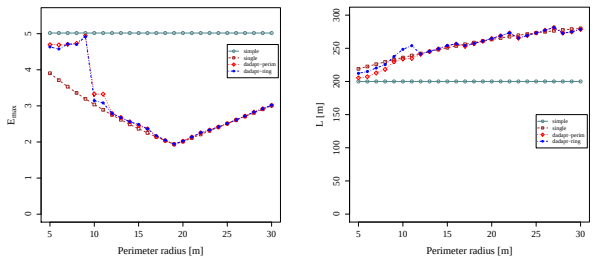
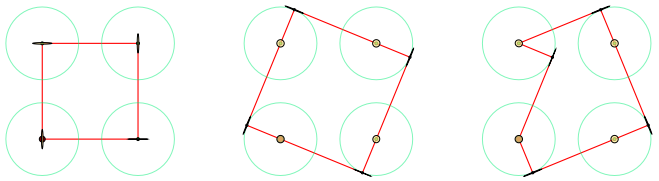


Fig. 3: Influence of the perimeter to the solution quality, *square* scenario, $\eta = 0.1, \tau = 0.1, \rho = 20$.



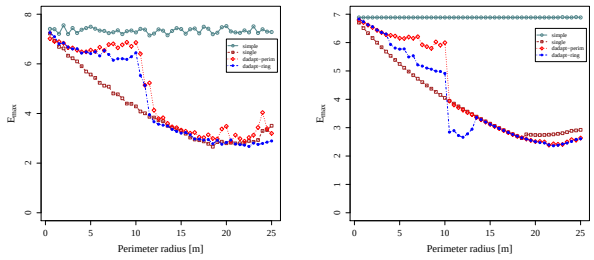
(a) *simple*, L=200 m, E_{max}=5 m

(b) *single-perim*, L=261 m, E_{max}=2 m

(c) *dadapt-ring*, L=261 m, E_{max}=2 m

Fig. 4: Solutions for the *square* scenario, perimeter at 19 m. The green disks are goals, the green circles denote perimeters, the found path is in red and ellipses denote localization uncertainty at goals (yellow) and perimeter points (blue).

times lower localization uncertainty at the goals, while the length of the path is about thirty percents longer. The proposed SOM algorithm with the dual adaptation provides competitive results to the *single-perim* variant in this simple scenario. The lowest E_{max}=1.94 m is achieved for the perimeter at 19 m. The variances of the computed quality metrics are very small due to simple configuration of the goals, the highest values are for the *dadapt* variants, but they are typically less than one meter for L and they are in hundredths for E_{max}.



(a) maximal localization uncertainty E_{max}

(b) E_{max} of the best found solution

Fig. 5: Influence of the perimeter to the solution quality, *square* scenario, $\eta = 0.1, \tau = 0.1, \rho = 20$.

To examine capabilities of the proposed algorithm in a larger environment fourteen goals have been placed within Charles's Square location. Experimental results are presented in Fig. 5. The *dadapt* algorithms provide worse average solutions up to perimeter around twelve meters. However,

consideration of the best found solution provides interesting observation. The *dadapt-ring* algorithm provides better solutions for perimeters between ten and fourteen meters. In these cases, the found solutions have localization uncertainty about one meter lower than for the *single-perim* variant. The perimeter at 18.5 m provides the lowest values of E_{max}. The best solution found by the *dadapt-ring* algorithm for additional twenty new runs and perimeter at twelve meters is depicted in Fig. 6. The best solution provided by the *single-perim* algorithm variant has quality metrics L=641.5 m with E_{max}=3.65 m.

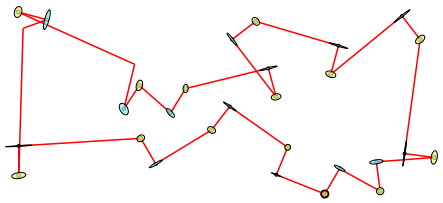


Fig. 6: The best found solution for *Charles's square* scenario, *dadapt-ring*,d_p=12 m,L=674.4 m, E_g=2.85 m.

B. Computational Requirements

The algorithms have been implemented in C++, compiled by the G++ 4.2 with the '-O2' optimization flag and executed at workstation with 2 GHz CPU. The *simple* and *single-perim* algorithms variants provide solution in units of milliseconds (including the solution of the TSP). The *dadapt* algorithm is more computationally intensive due the used number of perimeters points, particularly solutions for the *square* scenario have been found in tens of milliseconds and less than four hundred milliseconds in the case of *Charles's square* scenario. Even though the covariance matrix is computed for each node after each adaptation of the winner node, the simplicity of Equation (4) allows to plan several paths and select the best found solution or plan in real-time.

C. Real Indoor Experiments

Two scenarios have been used in real experiments within indoor environment. In these scenarios, a white color card with dimensions 85.6× 54 mm has been placed at each goal. The AR drone has been manually navigated along the given path, then it has been requested to traverse the closed path autonomously. The drone has been placed at the last goal approximately in the directions of the first goal or the perimeter point at each run. At each goal (over the card) the bottom camera has been used to take a snapshot of the card. The success of the navigation has been measured by the number of observed cards, i.e. a detected card in the snapshot over the goal area. The view angle of the vertical camera is approximately sixty degrees and the high of the drone has been in 1.0 - 1.5 m above the goal.

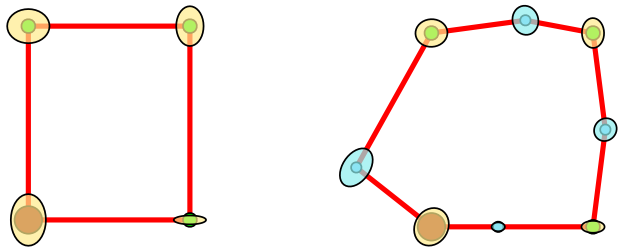
Scenario A - In the first scenario, goals form a rectangle with dimensions 6.25 × 3.75 m. Experimental results from five runs are presented in Table I. The *simple* path is a direct connection of the goals, while perimeter points have been manually set before each goal in the *perim* path.

TABLE I: Scenario A

Path Planning Method	Success Rates%				
	g_1	g_2	g_3	g_4	overall
simple	20	20	40	40	30
perim at 1.77m	60	60	40	20	45

Scenario B - A rectangle with 3.750 × 4.375 m and perimeter at 2.2 m has been used in this scenario. Beside the *simple* path, the best found solution by the method *dadapt-perim* has been used. Several snapshots of the goal area have been taken for the card detection, because of the following control issue. The images from the on-board cameras are transfered to the laptop for the image processing, i.e. SURF computation and landmark matching. The WiFi connection with the AR Drone caused unpredictable delays. These delays did not significantly affect the main navigation loop, however the command to take a snapshot of the goal area has been sometimes delayed. Even though the AR Drone tries to stabilize its position, under certain circumstances it is slightly moved. Thus, to avoid possible miss detection of the goal area from a single snapshot, several images have been captured and the goal visit has been considered as successful if the card has been detected in at least one snapshot.

The used paths are shown in Fig. 7. Notice that the perimeter point is needed only for the last goal g_4. The first perimeter point lies on the straight line segment from the start goal the g_1. In other cases the perimeter points are at distance 30 cm from the straight line segment, which in fact is very close to the expected localization precision. Success rates of the detected goals from tens runs are presented in Table II. The AR Drone during experiments is shown in Fig. 8.



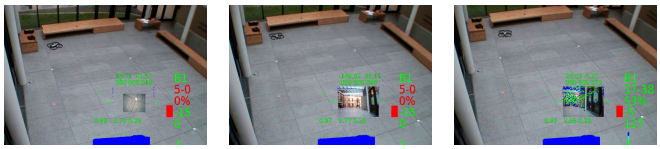
(a) *simple*, E_{max}=0.58m (b) *dadapt-perim*, E_{max}=0.45m

Fig. 7: Found paths by the *simple* and *dadapt-perim* algorithm variants and for the perimeter at 2.2 m in the scenario B.

The *single-perim* algorithm variant is not suitable for this scenario as distances between goals are relatively small and perimeter points are not necessary for first goals. The best found solutions has E_{max}=0.5 m and is about three meters longer than the *dadapt-perim* solution.

TABLE II: Scenario B

Path Planning Method	Success Rates%				
	g_1	g_2	g_3	g_4	overall
simple	100	100	60	70	82.5
dadapt-perim	100	100	90	90	95.0



(a) goal snapshot (b) front camera view (c) detected landmarks

Fig. 8: The AR Drone during experiment, the user interface overlays the real scene, the small picture is from the AR drone on-board cameras.

VI. Discussion and Conclusion

The presented results show that the expected localization uncertainty can be decreased by consideration of heuristic function describing evolution of the localization error. The used heuristic function is based on simple and stable navigation method, thus regarding to its real performance [7] and presented experimental results it seems that the function is computationally efficient and sufficiently informative. The proposed multi-goal path planning algorithm shows how the function can be used in the path planning task. Even though the results show uncertainty reduction, the benefit of the used SOM schema is mainly in providing solution of the multi-goal path planning problem. The straightforward *single-perim* variant provides better results in most cases with less required computational times. The efficiency of perimeter point placement by the *single-perim* method is caused by the dominant longitudinal error, because the right angle constraint is sufficient to decrease this type of error. In a general case, consideration of other orientation of the error, i.e. axes of the error ellipse, can lead to higher reduction. Also the axes directions depend on the order of goals in the path, thus the post-processing procedure can provide sub-optimal solutions. From this point of view, the best found solution found by the *dadapt* algorithm can be a motivation for further investigation of SOM application in the formulated multi-goal path planning problem. Moreover, for the small indoor environment, the *dadapt* method is able to find solutions in which all perimeter points are not needed. The expected benefit of SOM is in its flexibility to deal with various problem variants, e.g. considering obstacles in the environment [13], restrictions of directions to reach goals or in the case of several mobile robots [14].

Our future intention is to examined the proposed method in a real-world outdoor scenario[1]. The main idea of the uncertainty reduction is to increase the reliability of the inspection. The found plan represents only expected localization uncertainty, thus it is clear that real performance will be different and a robot will unlikely be navigated precisely to the goal position. On the other side, it is sufficient if the robot is navigated to the goal vicinity, because a robot can be then locally navigated to the goal by another navigation method. For an UAV, it means that forward looking camera can be used for the navigation to the goal vicinity, while the vertical camera is used for the local navigation to the goal. The execution of the plan has one important aspect relating to the localization uncertainty. If the goal is successfully recognized and its position is known, it can be used to localize the robot. Thus, additional reduction of the localization error is achieved, which increases the probability of the next goal visit. These ideas will be investigated in our future work.

[1]We have planed experiment in Charles Square park, however due to weather conditions and low turbine power of the UAV it was not possible. Even small wind made the robot control unpredictable.

Acknowledgments

This work was supported by the Grant Agency of the Czech Technical University in Prague, grants No. SGS10/185 and No. SGS10/195 and by the Ministry of Education of the Czech Republic under program "National research program II" by the projects 2C06005 and by the project No. 7E08006. The support of EU under the project ICT-2007-1-216240 is also acknowledged. We would also like to thank the Parrot company for providing AR drones to us.

References

[1] A. Remazeilles and F. Chaumette, "Image-based robot navigation from an image memory," *Robot. Auton. Syst.*, vol. 55, no. 4, pp. 345–356, 2007.

[2] S. Segvic, A. Remazeilles, A. Diosi, and F. Chaumette, "A mapping and localization framework for scalable appearance-based navigation," *CVIU*, vol. 113, no. 2, pp. 172–187, February 2009.

[3] A. Censi, D. Calisi, A. D. Luca, and G. Oriolo, "A Bayesian framework for optimal motion planning with uncertainty," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Pasadena, CA, May 2008.

[4] U. Frese, "A discussion of simultaneous localization and mapping," *Auton. Robots*, vol. 20, no. 1, pp. 25–42, 2006.

[5] A. Martinelli, N. Tomatis, and R. Siegwart, "Some results on SLAM and the closing the loop problem," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Canada, 2005, pp. 334–339.

[6] S. J. Julier and J. K. Uhlmann, "A counter example to the theory of simultaneous localization and map building," in *IEEE/RSJ Int. Conf. Robotics and Automation*, 2001, pp. 4238–4243.

[7] T. Krajník, J. Faigl, V. Vonásek, K. Košnar, M. Kulich, and L. Přeučil, "Simple yet stable bearing-only navigation," *Journal of Field Robotics*, vol. 27, no. 5, pp. 511–533, 2010.

[8] Z. Chen and S. T. Birchfield, "Qualitative vision-based path following," *Trans. Rob.*, vol. 25, no. 3, pp. 749–754, 2009.

[9] S. Somhom, A. Modares, and T. Enkawa, "A self-organising model for the travelling salesman problem," *Journal of the Operational Research Society*, pp. 919–928, 1997.

[10] M. Saha, T. Roughgarden, J.-C. Latombe, and G. Sánchez-Ante, "Planning Tours of Robotic Arms among Partitioned Goals," *Int. J. Rob. Res.*, vol. 25, no. 3, pp. 207–223, 2006.

[11] H. Bay, T. Tuytelaars, and L. Gool, "SURF: Speeded up robust features," in *Proc. 9th European Conf. Comput. Vision*, Graz, Austria, 2006, May.

[12] "Parrot AR Drone," http://ardrone.parrot.com/parrot-ar-drone/en, cited July 2010.

[13] J. Faigl, M. Kulich, V. Vonásek, and L. Přeučil, "An Application of Self-Organizing Map in the non-Euclidean Traveling Saleman Problem," *Neurocomputing*, to appear.

[14] J. Faigl, "Multi-goal path planning for cooperative sensing," Ph.D. dissertation, Czech Technical University in Prague, 2010.