

Autonomous exploration with prediction of the quality of vision-based localization

Hélène Roggeman* Julien Marzat*
Anthelme Bernard-Brunel* Guy Le Besnerais*

* ONERA – The French Aerospace Lab, F-91123, Palaiseau, France.
e-mail: *firstname.lastname@onera.fr*.

Abstract: This paper presents an algorithm to perform autonomous exploration with robotic platforms equipped with a stereo-vision system in indoor, unknown and cluttered environments. The accuracy of the vision-based localization depends on the quantity of visible features in the scene captured by the cameras. A Model Predictive Control approach permits to perform the exploration task with obstacle avoidance and taking into account the quality of the scene in order to avoid areas where the visual odometry is likely to fail. Experiments were carried out with a mobile robot to assess the improvement in localization accuracy and coverage for exploration.

Keywords: Autonomous vehicles, Autonomous exploration, Stereo-vision, Model Predictive Control

1. INTRODUCTION

Autonomous exploration allows to build maps of unknown environments without human intervention. This can be interesting, for instance, for search-and-rescue missions in dangerous areas. UAVs and mobile robots can be complementary to perform this type of missions. Their success depends on the accurate localization of the robot. The localization algorithms are designed to be embedded on UAVs and mobile robots. The UAVs have a low payload that implies a limitation of the number of embedded sensors. Moreover, these exploration missions are most of the time conducted in indoor environments, where no global localization systems, such as GPS, are available. Thus, a good solution to address these constraints is to install camera sensors, because they are lightweight, inexpensive and the images provide a high amount of information. A stereo rig composed by two cameras with a visual odometry algorithm allows to compute the localization of the platform from features extracted in the images (Sanfourche et al., 2013).

All passive vision-based navigation systems are likely to fail in low-textured environments, where the lack of interest points in the images prevent the computation of a good localization. The aim of this work is to develop an autonomous exploration system with a command strategy that seeks to avoid the situations where the robot is likely to lose its localization due to lack of texture in the environment. We experiment with a mobile robot in order to validate the developed strategy, but the overall system can easily be adapted for a UAV.

1.1 Related Work

The issue of evaluating the scene quality for vision-based localization was studied in the context of navigation between waypoints. Sadat et al. (2014) take into account the richness of the environment for the path planning,

they used a RRT* algorithm and add a criterion which computes a viewpoint score based on the density of the triangle in the 3D mesh of the environment. Mostegel et al. (2014) evaluate the quality of the camera motions for the localization quality and the possibility of seeing new features. A combination of criteria on the features is computed and used to define if a future position will be well suited for the localization. In these two references, the robotic platforms are equipped with a monocular camera, which involves specific depth estimation issues to compute the localization. In previous work (Roggeman et al., 2016), we focused on the same problematic but with stereo-vision system which makes it possible to obtain directly information about the depth of the points. The present paper elaborates on this work by addressing autonomous exploration missions.

Some authors were interested in the active reduction of the uncertainty during an autonomous exploration mission: Bourgaul et al. (2002) aimed at exploring and building an accurate map of the environment with a mobile robot equipped with a laser range finder selecting the control actions that maximize the accuracy of the localization. Bryson and Sukkarieh (2008) developed an information-based path planning method for a UAV. It plans a trajectory which maximizes the accuracy of the map and the vehicle location during the exploration of unknown areas. It is based on the computation of the entropic information gain before and after taking an action. This system can be used with vision sensors.

1.2 Problem statement

This paper describes a complete architecture for autonomous exploration on robotic platforms. The mission considered in this paper is the exploration by a mobile robot of an unknown and cluttered environment which presents some low textured areas. The robot has to complete its exploration mission avoiding the obstacles in the

room and keeping an accurate localization during all the mission.

To achieve this mission, it is necessary to localize the robot and map the environment. The localization is estimated from the images of the stereo-vision system by the visual odometry algorithm (Sanfourche et al., 2013), described in Section 2.1. The mapping task is possible thanks to a RGB-D sensor, explanations can be found in Section 2.2. For the detection of the low-textured areas, a criterion based on the prediction of the amount of information available in the future images permits to define if a position is appropriate for the localization. This criterion is explained in Section 3. The Model Predictive Control (MPC) strategy presented in Section 4, uses the information of mapping, localization and quality of texture of the scene in order to find the optimal control to send to the motors. Experiments were made in real situations and the results are reported in Section 5.

2. PERCEPTION

2.1 Visual odometry algorithm: eVO

The localization of the robot is ensured by a visual odometry algorithm, using the images providing by the stereo rig. The aim is to estimate the localization of the robot from its starting point. In our experiments, we use eVO (Sanfourche et al., 2013) but other algorithms could be used as well (Klein and Murray, 2007).

The following is a brief description of the algorithm. Two tasks are working in parallel:

- Mapping: this task consists in providing a map with a limited number of points localized in space. Interest points (Harris and Stephens (1988) or FAST Rosten and Drummond (2006)) are extracted from both images and matched. The 3D position of the points in space is then computed by a triangulation (see Eq. 1).
- Localization: The matching between the 2D points in the left image and the 3D points in the map derives from the temporal tracking of 2D points with KLT (Shi and Tomasi, 1994). The position and orientation of the left camera are computed by minimization of the reprojection error, within a RANSAC procedure (Fischler and Bolles, 1981).

The 3D points computed during the mapping task will serve to evaluate the ability of the robot to localize itself from a given position, see Section 3.

2.2 Environment reconstruction

For the obstacle avoidance and the exploration tasks, it is necessary to have an occupancy map of the environment. A Kinect sensor is installed on the robot and gives a 3D representation of the environment in a 3D point cloud format. We first remove the ground plan, using a RANSAC method. Then, the obtained point cloud is transformed into an Octomap model (Hornung et al. (2013)), which is a representation of the volumetric occupancy. Finally, the Octomap is projected onto the ground plane and two 2D maps are created: an exploration map with the explored and unexplored areas and an obstacle map with

the occupied and free areas. At time t_n , the exploration map is represented as a matrix denoted $G(n)$ whose elements $g_{i,j}$ are 0 or 1: 0 if the location is not explored, 1 if it is explored.

3. VISUAL QUALITY FOR THE LOCALIZATION

From the 3D points extracted by the odometry algorithm and with a known future position, the proposed visual quality criterion is derived from the prediction of the number of visible points in the future images, considering the uncertainty (Roggeman et al., 2016). The whole process is illustrated in Figure 1 and described in the following.

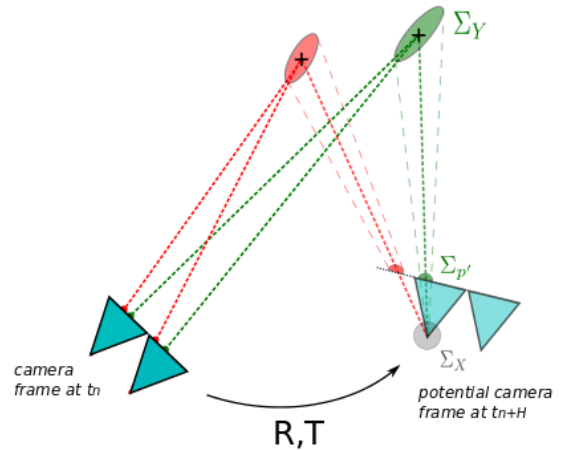


Fig. 1. Two 3D points are triangulated with their covariances at time t_n . The two points are projected onto the camera plane after a displacement (R, T) : the green one is predicted to lie in the image whereas the red one is outside.

3.1 Future point position

From the 2D points extracted in the stereo images, the position of a 3D point in the current camera frame, defined by the position of the left camera, is given by

$$Y = (x, y, z)^T = \Pi^{-1}(u, v, d) = \frac{-b}{d} \cdot \begin{pmatrix} u - u_0 \\ v - v_0 \\ \alpha \end{pmatrix} \quad (1)$$

(u, v) is the 2D position of the point in the image and d is the disparity. b denotes the baseline between the left and right camera, α , the focal length and (u_0, v_0) are the coordinates of the principal point.

A change of basis is necessary in order to express the position of the 3D point in the future camera frame.

$$\tilde{Y}' = P \cdot \tilde{Y} = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} \cdot (x, y, z, 1)^T \quad (2)$$

R and T are respectively the rotation and translation between the two frames.

The 3D point is then projected on the future image plane, corresponding to the camera frame.

$$\begin{aligned} p &= \Pi(Y) \\ \tilde{p} &= (u, v, 1)^T = z^{-1} K \cdot Y \end{aligned} \quad (3)$$

with K , the camera matrix

$$K = \begin{pmatrix} \alpha & 0 & u_0 \\ 0 & \alpha & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4)$$

The expression of p' the position of the future 2D point as a function of the position p , the disparity d and the displacement parameters Θ is:

$$p' = f(\Theta, u, v, d) \quad (5)$$

$$p' = \Pi[Y'] = \Pi[P \cdot \tilde{Y}] = \Pi[P \cdot \tilde{\Pi}^{-1}(u, v, d)] \quad (6)$$

3.2 Uncertainty

The uncertainty on the position of the point p' is expressed as a 2D covariance matrix. We assume that the uncertainty related to Θ and (u, v, d) are independent. The covariance is:

$$\Sigma_{p'} = J_{f_\Theta} \cdot \Sigma_\Theta \cdot J_{f_\Theta}^T + J_{f_{u,v,d}} \cdot \Sigma_{u,v,d} \cdot J_{f_{u,v,d}}^T \quad (7)$$

with J_{f_Θ} and $J_{f_{u,v,d}}$ the Jacobian matrices of f with respect to Θ and (u, v, d) , respectively.

$$J_{f_\Theta} = \frac{\partial f}{\partial \Theta} = J_\Pi(Y') \cdot J_{Y'}(Y) \quad (8)$$

$$J_{f_{u,v,d}} = \frac{\partial f}{\partial u, v, d} = J_\Pi(Y') \cdot R \cdot J_\Pi(u, v, d) \quad (9)$$

J_Π , $J_{\Pi^{-1}}$ and $J_{Y'}$ are the Jacobian matrices of the projection function, the triangulation function and the change of basis function. Their expressions can be found by derivating the functions given in Section 3.1.

3.3 Prediction of the number of points

We want to estimate the probability that the point p' lies in the future image. Using the 2D covariance matrix $\Sigma_{p'}$, representing the uncertainty on the position of the predicted point p' , the corresponding 90% confidence ellipse is considered. $x = (u, v)^T$ is an image point. The ellipse equation in the image frame is

$$(x - p')^T \Sigma_{p'}^{-1} (x - p') = s \quad (10)$$

with $s = 4.605$ for a 90% confidence ellipse, this value can be found in a table of χ^2 distribution with 2 degrees of freedom.

Then, the area of the intersection between the ellipse and the image is computed. If the ellipse is entirely located in the image, the area is that of the ellipse, if it is completely out, the area is zero. In the other cases, the area is estimated by computing the double integration of the domain delimited by the ellipse and the edges of the image.

The probability of a point to lie into the future image is estimated by dividing the obtained area with the whole ellipse area. A threshold on this probability is used to define if a point is considered into the image. The final criterion is the number of points satisfying this condition.

4. CONTROL

4.1 Robot kinematic model

We denote by $X = (x, y, \theta)$, the 2D position and the orientation of the robot, $U = (v, \omega)$ is the control input, with v the linear speed and ω the angular speed and t_e is the sampling period. The kinematic model of the mobile robot is:

$$\begin{cases} x_{n+1} = x_n + t_e v_n \cos \theta_n \\ y_{n+1} = y_n + t_e v_n \sin \theta_n \\ \theta_{n+1} = \theta_n + t_e \omega_n \end{cases} \quad (11)$$

4.2 MPC

Using the robot model (11), some trajectories are simulated within a finite horizon H_p , from a sequence of control inputs \mathcal{U} , which are defined on a control horizon H_c . The control inputs and simulated state sequences at time t_n are denoted:

$$\mathcal{U}_n = \{U_n, U_{n+1}, \dots, U_{n+H_c-1}\} \quad (12)$$

$$\mathcal{X}_n = \{X_{n+1}, X_{n+2}, \dots, X_{n+H_p}\} \quad (13)$$

The linear and angular velocities are bounded by $(-v_{\max}, -\omega_{\max})^T$ and $(v_{\max}, \omega_{\max})^T$.

A cost function $J(\mathcal{U}_n, \mathcal{X}_n)$ is defined to mathematically represent the mission objectives (see Section 4.3). The minimization of this cost function gives the optimal control \mathcal{U}_n^* to apply to the system as:

$$\begin{aligned} \mathcal{U}_n^* &= \arg \min_{\mathcal{U}_n} J(\mathcal{U}_n, \mathcal{X}_n) \\ &\text{with } X_k \text{ satisfying (11),} \\ &\forall k \in [n+1, n+H_p] \end{aligned} \quad (14)$$

Only the first component U_n^* of the optimal solution is applied to the system and the process is repeated at the next time step.

This optimization problem is non-linear and non-convex. There are different solutions to resolve it. Global optimization algorithms can be used but their execution time is not constant and it can be too long for a real-time application. In this work, we chose to define a finite set of predefined control sequences and select the optimal solution in this set. This strategy permits to limit the risk of falling in a local minimum.

4.3 Cost function

The expression of the cost function is

$$J = w_{\text{loc}} J_{\text{loc}} + w_{\text{expl}} J_{\text{expl}} + w_{\text{obs}} J_{\text{obs}} \quad (15)$$

Each cost J_\bullet represents an objective of the mission. They are explained in the following paragraphs. All the costs are normalized. The weights w_\bullet give more importance to one objective compared to the others. The modification of the weights induces changes in the behavior of the robot and it requires trial and error to find an optimal tuning of the weights. This is discussed in Section 5.2.

Localization quality cost J_{loc} This cost relies on the visual quality criterion defined in Section 3. It favors trajectories which go through positions where a large number of points will be visible.

$$J_{\text{loc}} = 1 - \frac{N(X_{n+H_{\text{loc}}}, \mathcal{J}_n)}{N_{\text{max}}} \quad (16)$$

where $N(X_{n+H_{loc}}, \mathcal{Y}_n)$ is the number of predicted landmarks at $n + H_{loc}$ and N_{max} is the total number of 3D points. H_{loc} is the time horizon where the prediction is computed, it must be fixed beforehand (see Section 5.2).

Exploration cost J_{expl} This cost favors trajectories which go towards the unexplored areas. From a copy of the exploration map at the current time $G(n)$ (see Section 2.2), the amount of future explored space is estimated on the trajectory. From the known values of the range and the opening angle of the mapping sensor, for each predicted position, the corresponding explored space is added as explored area in $G(n)$. The obtained map, after the time horizon, is denoted $G(n + H_p)$. The proposed cost tends to maximize the additional explored area from $G(n)$ to $G(n + H_p)$. The same process was used successfully in a previous work (Roggeman et al., 2014). The expression of the cost is:

$$J_{expl} = \frac{r^2}{H_p \cdot \alpha \cdot d^2} \sum_i \sum_j g_{i,j}(n) - g_{i,j}(n + H_p) \quad (17)$$

d is the range and α the opening angle of the sensor, r is the resolution of the map. i and j are the coordinates of the elements of G .

Obstacle avoidance cost J_{obs} This cost penalizes trajectories which go too close from the obstacles or which intersect them. The expression of the cost is:

$$J_{obs} = \frac{1}{H_p} \sum_{k=n+1}^{n+H_p} f_{obs}(G_{dist}(X_k)). \quad (18)$$

The obstacle map (see Section 2.2) is transformed into a distance map, called G_{dist} : for each square in the grid, the distance to the nearest obstacle is computed.

f_{obs} is a function which normalizes the distance into a cost between 0 and 1. Two distances are defined: d_{sec} is the security distance, the robot must not exceed this limit and d_{des} is a distance from which the obstacle is no longer considered. $f_{obs}(d)$ is equal to 1 if $d < d_{sec}$ and to zero if $d > d_{des}$, with a smooth decrease in-between. The curve is displayed in Figure 2.

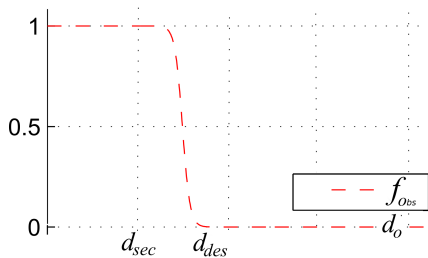


Fig. 2. Penalty function for obstacle avoidance

5. EXPERIMENTS

5.1 Experimental set-up

Mission The aim of the mission is to explore the room showed in Figure 3. It measures approximately 11x11 meters. In this room, there is a wall with low texture. If the robot arrives in front of this wall, the visual localization

is likely to fail. We want to verify if the addition of the criterion on the quality of the localization permits to improve the behavior of the robot, that is, whether it succeeds to explore the whole space while keeping an accurate localization. Moreover, there are two obstacles in the room that the robot has to avoid.



Fig. 3. The robot in its environment, in the back of the room, there is a textureless wall which is a difficulty for the visual localization

Material For the experiments, we use a turtlebot (see Figure 4), equipped with a stereo rig composed of two cameras with 4.0mm lens and separated by a 21cm long baseline and a Kinect sensor, with a range fixed to 2.5 m and an opening angle of 0.5 rad. The perception algorithms are working on the embedded computer whereas the MPC algorithm is working on a deported station. A motion capture system computes the localization of the robot, it is considered as the ground truth and serves for comparison with the visual localization.

Parametrization

- Localization criterion
 - Probability threshold: $s_{proba} = 0.5$
 - Uncertainty $\Sigma_{\Theta} = \text{diag}(\sigma_{\theta_x}^2, \sigma_{\theta_y}^2, \sigma_{\theta_z}^2, \sigma_x^2, \sigma_y^2, \sigma_z^2)$ and $\Sigma_{u,v,d} = \text{diag}(\sigma_u^2, \sigma_v^2, \sigma_d^2)$ with:
 - position: $\sigma_x = \sigma_y = \sigma_z = 0.005$ m
 - orientation: $\sigma_{\theta_x} = \sigma_{\theta_y} = \sigma_{\theta_z} = 0.001$ rad
 - image point position: $\sigma_u = \sigma_v = 0.2$ pixel
 - disparity: $\sigma_d = 0.4$ pixel
- MPC
 - Time step: $t_e = 0.25$ s
 - Horizons: $H_p = 20$, $H_c = 8$
 - Obstacle distances: $d_{des} = 0.5$ m, $d_{sec} = 0.5$ m
 - Map resolution: $r = 0.2$ m per pixel
 - Speed limitations: $v_{max} = 0.25$ m.s⁻¹, $\omega_{max} = 0.4$ rad.s⁻¹
 - Weights: $w_{expl} = 0.3$, $w_{obs} = 0.7$

The weights on the exploration cost and the obstacle cost are fixed at the beginning. Only the weight on the quality of the localization is modified during the experiments. The weights are normalized after each change on w_{loc} so that the sum of the weights remains equal to 1.

5.2 Results

In Figure 5, two results of exploration missions are displayed. In the left image, the weight on the localization quality cost is zero. At the beginning, the robot turns to the right, it is the optimal trajectory because it maximizes

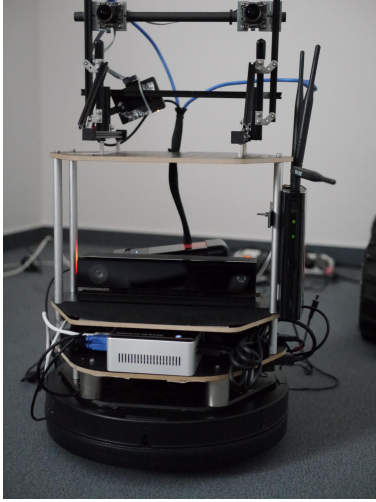


Fig. 4. The Turtlebot platform used for the experiments: the stereorig is visible on the upper part; and the Kinect sensor is below.

the observed area, but this leads the robot in front of the textureless wall and the visual localization becomes inaccurate, rapidly leading to a large drift of the estimated trajectory and the exploration mission cannot be performed. In the second experiment, the weight w_{loc} is increased to 0.07. The robot begins at the same position, but in this case, it goes straight forward and does not face the wall. The mission then continues successfully, the zone is entirely explored.

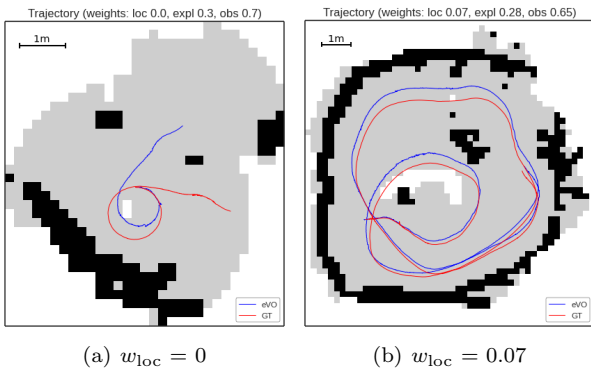


Fig. 5. Results of two experiments with different values of w_{loc} , the visual localization is the blue line and the ground truth is the red one, the grey area is the free explored spaces and the black area is the occupied areas

Others experiments were made to confirm these results and to define what could be the best parametrization for the robot in order to have the best exploration result. The results of these experiments are summarized in Table 1. The values of H_{loc} and w_{loc} were modified, and we computed measures on the localization error and on the success of the exploration mission. The relative error is the error between the total travelled distance computed by the visual localization algorithm and the ground truth relative to the total travelled distance. This value is expressed as a percentage. The maximal distance column shows the maximal distance estimated by visual localization between two time steps in the visual localization. As the maximal speed was set

nb	H_{loc}	w_{loc}	duration	distance	coverage	rel. err.	max dist.
0	0	0.0	66.41	19.03	27.24	107.98	0.513
1	0	0.0	16.36	2.92	13.28	40.19	0.073
2	0	0.0	14.1	2.74	12.24	29.97	0.08
3	0	0.0	32.76	6.26	26.84	21.39	0.111
4	0	0.0	35.09	7.58	31	91.42	0.152
5	1	0.02	184.67	52.07	72.92	25.76	0.063
6	1	0.05	229.18	63.82	75.04	34.96	0.076
7	1	0.07	186.51	48.75	71.68	30.62	0.08
8	1	0.11	66.54	19.1	51.68	33.48	0.068
9	2	0.02	102.95	28.75	61.12	28.18	0.065
10	2	0.05	132.74	37.37	66.04	37.39	0.058
11	2	0.07	98.78	27.37	59.8	21.33	0.064
12	2	0.11	91.7	23.31	55.96	24.83	0.063
13	3	0.07	106.68	28.13	60.72	32.63	0.064
14	3	0.08	106.53	27.53	59.08	23.26	0.071
15	4	0.07	80.65	20.36	53.96	19.5	0.059

Table 1. Table of the experimental results: the duration is expressed in seconds, the distance in meters, the coverage as a percentage of the whole area to explore, the relative error as a percentage and the maximal distance in meters.

to 0.25 m.s^{-1} and the maximal duration between two time steps is approximately 0.20 s, the maximal travelled distance between two time steps is approximately 0.05 m. Hence the value of the maximal distance allows to evaluate to the drift induced by the visual navigation.

The first five runs (number 0-4) were done without using the localization cost (ie. $H_{loc} = w_{loc} = 0$). Among them, three experiments present a high error on the maximum distance (experiences 0, 3 and 4 with 0.513, 0.111 and 0.152 m maximum distance respectively), whereas with a positive weight, the maximal error is below 0.08 m. These experiments also present the lowest exploration coverages because they were too hazardous and had to be stopped by the user. We can conclude that, the addition of the criterion on the quality of the localization permits to avoid the occurrence of large drift in the visual localization.

During the experiments (5-15), H_{loc} was fixed between 1 and 4, and Table 1 shows that the coverage is higher if H_{loc} is lower. Indeed, if H_{loc} is low, the robot goes more easily in the corners of the room whereas it tends to avoid them when H_{loc} is higher (see Figure 6). In this figure, with the same value of w_{loc} , the result of the exploration appears completely different depending on the value of H_{loc} . Let us recall that the visual quality is computed from 3D point seen in the past. The best way to keep them in the field of view within a large horizon of time is to follow a straight trajectory. Using a shorter horizon permits curved paths and eases the exploration of corners.

Figure 7 present four results obtained with a short horizon $H_{loc} = 1$ and various values of w_{loc} . We can see that the results relative to the exploration depends on the value of w_{loc} : if w_{loc} is too high (experience 8, $w_{loc} = 0.11$), the robot adopts a cautious behavior by keeping a large number of previously seen point in its field of view, which precludes him to explore new areas whereas with a lower value (experiences 5, 6 and 7, $w_{loc} = 0.02, 0.05$ and 0.07

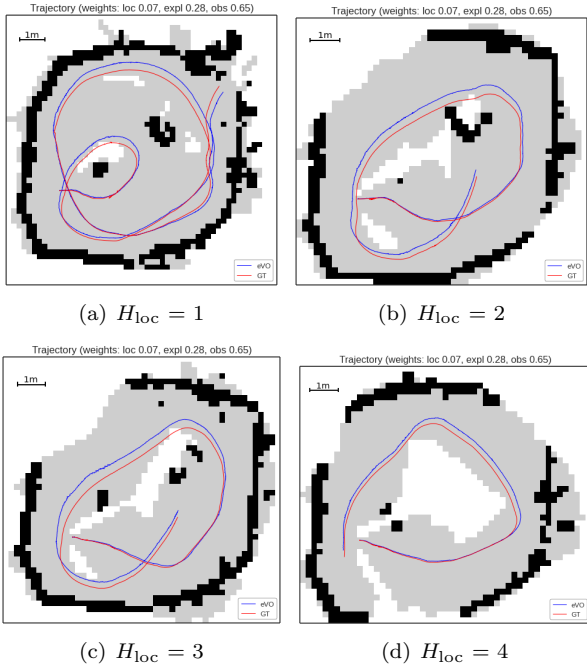


Fig. 6. Results of four experiments with different values of H_{loc} with $w_{loc} = 0.07$

respectively), the robot explores the whole area, keeping a good visual localization.

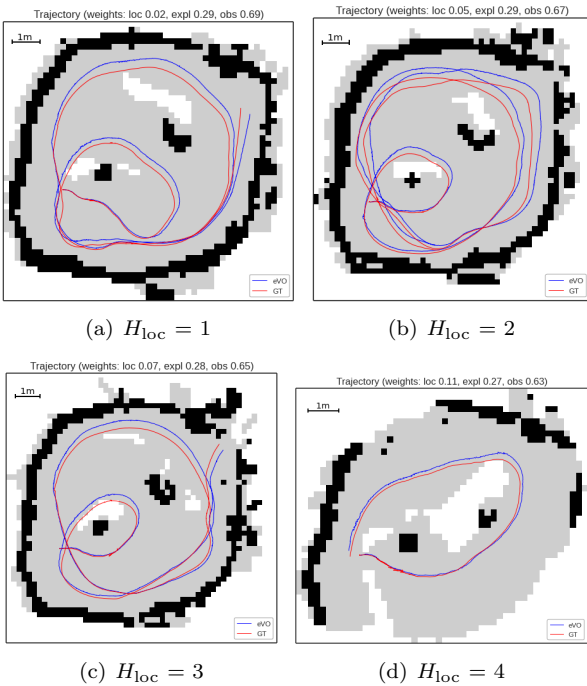


Fig. 7. Results of four experiments with different values of w_{loc} with $H_{loc} = 1$

According to the previous parametric experimental study, we conclude that a good parametrization is using $H_{loc} = 1$ and w_{loc} between 0.02 and 0.07.

6. CONCLUSION

In this paper, we have presented a complete architecture to perform autonomous exploration with a mobile robot

using a vision-based localization system. The model predictive control strategy is designed to avoid the situations where the localization is likely to fail due to the lack of texture in the environment. Experiments were carried out in the real world and demonstrated that this system can improve the exploration behavior of the robot and ensure safe localization.

REFERENCES

- Bourgaul, F. et al. (2002). Information based adaptive robotic exploration. In *Proceedings of the IEEE/RSJ International Conference on Robots and Systems, Lausanne, Switzerland*, volume 1, 540–545.
- Bryson, M. and Sukkariéh, S. (2008). Observability analysis and active control for airborne SLAM. *IEEE Transactions on Aerospace and Electronic Systems*, 44(1), 261–280.
- Fischler, M.A. and Bolles, R.C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395.
- Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, 147–151.
- Hornung, A. et al. (2013). OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3), 189–206.
- Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In *Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan*, 225–234.
- Mostegel, C. et al. (2014). Active monocular localization: Towards autonomous monocular exploration for multirotor MAVs. In *Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China*, 3848–3855.
- Roggeman, H., Marzat, J., Bernard-Brunel, A., and Le Besnerais, G. (2016). Prediction of the scene quality for stereo vision-based autonomous navigation. In *9th IFAC Symposium on Intelligent Autonomous Vehicles IAV*, 94–99. Leipzig, Germany.
- Roggeman, H., Marzat, J., Sanfourche, M., and Plyer, A. (2014). Embedded vision-based localization and model predictive control for autonomous exploration. In *IROS Workshop on Visual Control of Mobile Robots (ViCoMoR)*, 13–20. Chicago, United States.
- Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In *Proceedings of the European Conference on Computer Vision, Graz, Austria*, volume 1, 430–443.
- Sadat, S.A. et al. (2014). Feature-rich path planning for robust navigation of MAVs with mono-SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China*, 3870–3875.
- Sanfourche, M. et al. (2013). evo: A realtime embedded stereo odometry for MAV applications. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan*, 2107–2114.
- Shi, J. and Tomasi, C. (1994). Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Jerusalem, Israel*, 593–600.