2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Air-Ground Localization and Map Augmentation Using Monocular Dense Reconstruction

Christian Forster, Matia Pizzoli, Davide Scaramuzza

Abstract— **We propose a new method for the localization of a Micro Aerial Vehicle (MAV) with respect to a ground robot. We solve the problem of registering the 3D maps computed by the robots using different sensors: a dense 3D reconstruction from the MAV monocular camera is aligned with the map computed from the depth sensor on the ground robot. Once aligned, the dense reconstruction from the MAV is used to augment the map computed by the ground robot, by extending it with the information conveyed by the aerial views. The overall approach is novel, as it builds on recent developments in live dense reconstruction from moving cameras to address the problem of air-ground localization. The core of our contribution is constituted by a novel algorithm integrating dense reconstructions from monocular views, Monte Carlo localization, and an iterative pose refinement. In spite of the radically different vantage points from which the maps are acquired, the proposed method achieves high accuracy whereas appearance-based, state-of-the-art approaches fail. Experimental validation in indoor and outdoor scenarios reported an accuracy in position estimation of 0.08 meters and real time performance. This demonstrates that our new approach effectively overcomes the limitations imposed by the difference in sensors and vantage points that negatively affect previous techniques relying on matching visual features.**

I. Introduction

A heterogeneous robotic system consistent of both, ground and aerial robots of different sizes, shapes and with different sense-act capabilities could greatly assist professional rescuers in a search and rescue scenario. However, it is difficult for the same human operator to concurrently monitor and navigate multiple robots while coordinating with other operators. Therefore, the necessary technologies must be developed to allow heterogeneous robots to autonomously localize and move with respect to each other and thereby ease the task of the operator and provide the best possible situation awareness.

In this work we consider a single MAV that acts as a "flying external eye" for a ground robot. The MAV operates in close range to the ground robot and offers the ability to hover and move in complex three dimensional space and observe the scene from a vantage point inaccessible to the ground robot (see Figure 1). The use of very small and lightweight MAVs reduces safety concerns, costs, and increases the agility of the platform. However, active ranging devices such as laser rangefinders or RGBD sensors cannot currently be used due to payload and power consumption restrictions. The ground robot, on the other hand, can carry more payload such as active depth sensors, processors and may be equipped with a manipulator arm. The usefulness of such a heterogeneous robot team in a disaster scenario has recently been demonstrated in [1].

The authors are with the Artificial Intelligence Lab—Robotics and Perception Group, University of Zurich, Switzerland—http://rpg.ifi.uzh.ch. This research was partly supported by the Swiss National Science Foundation through project number 200021-143607 ("Swarm of Flying Cameras"), the National Centre of Competence in Research Robotics, and the CTI project number 14652.1.

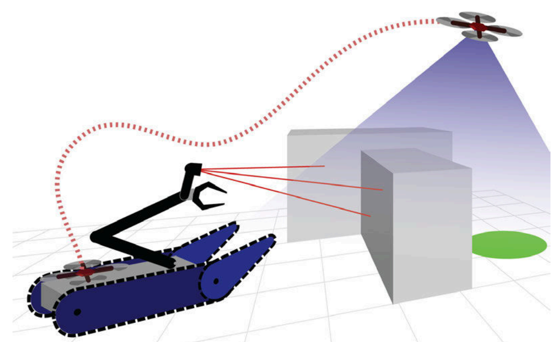


Fig. 1: The flying robot operating in close range to the ground robot provides a different vantage point for human tele-operators in a search-and-rescue scenario. We address the problem of autonomously localizing the aerial robot with respect to the ground-robot based on the structure of the scene.

In this paper, we address the problem of localizing the MAV with respect to the ground robot in close range. This capability will allow the robots to execute collaborative tasks and to present the teleoperator with a ground map which is augmented with aerial views from the MAV.

Due to payload restrictions, the MAV is equipped with a single downward-looking camera. On the other hand, the ground robot has a range sensor (either a laser or an RGBD camera) and further carries the main processing unit. Our experimental platforms are depicted in Figure 16.

Given the available sensory capabilities, there are two possible strategies to mutually localize the robots: (i) by leveraging relative observations between the MAV and the ground robot [2], (ii) or by matching and aligning maps computed by the MAV and the ground robot. The second option offers the advantage that the robots do not need to remain in the field of view of each other. However, the main challenge in the second strategy is the drastically different view points of the two robots (see Figure 2).

In this paper we propose a novel solution to this problem by leveraging the 3D surface computed from different view points and heterogeneous sensors. Through the alignment of both maps, the relative pose of the robots can be recovered. Computing a dense 3D surface from monocular cameras in real-time has only recently become feasible with the use of GPGPU computing [3], [4]. Therefore, we propose to distribute the processing between the robots. The MAV

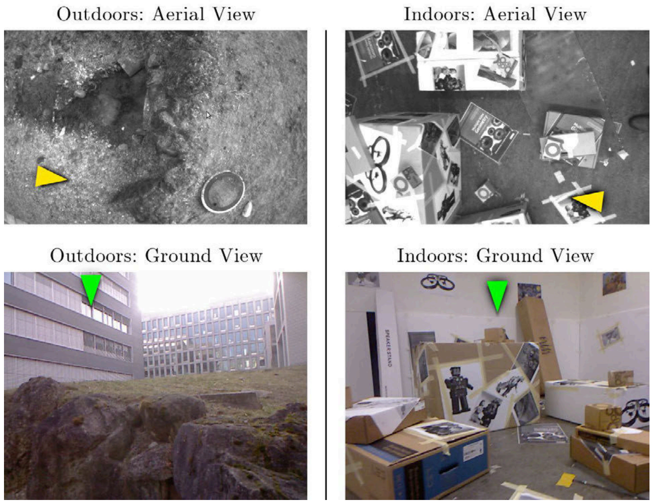


Fig. 2: Outdoor (left) and indoor (right) scenes observed from aerial and ground point of views. Robot poses are expressed by the arrows: yellow for the ground robot and green for the MAV.

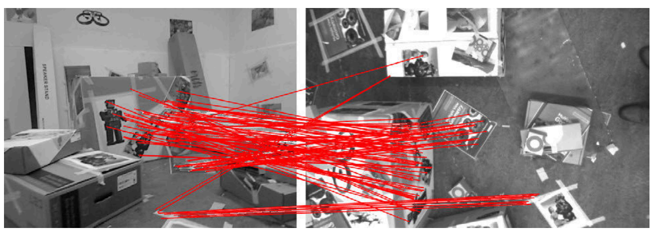


Fig. 3: Image feature matching results of the indoor scene using ASIFT [6]. Matches were found on planar textured surfaces. No matches were found in the outdoor scene depicted on the left of Figure 2.

computes its relative motion onboard using the downward looking camera [5] and, additionally, streams the video to the ground robot where a dense 3D model is computed and aligned with the ground robot's 3D map. Thereby, the relative pose of the robots is recovered.

We propose a solution for global alignment of the aerial and ground maps based on Monte Carlo Localization on the height-maps. Subsequently, the estimated transformation is refined through an Iterative Closest Point (ICP) algorithm. In experimental results we show that in a cluttered environment with sufficient 3D structure, we can compute the relative position with a precision of 0.078 m. Furthermore, we illustrate how the ground-based map can be augmented with the aerial view.

The outline of the paper is the following. In Section II we compare our approach to related works in the literature, while Section III provides an overview of the proposed method. In Section IV we present the SLAM methods operating on the MAV and the ground robot, while in Section V the dense reconstruction method is detailed. In Section VI, our Monte Carlo approach to global localization is described and in Section VII we present the iterative pose refinement. Section VIII reports about the experimental validation and in Section IX the conclusion is drawn.

II. Related Work

Very little research has addressed close-range relative localization of aerial and ground-robots. In most related works, the aerial robot flies outdoors higher than 20 meters, and can be localized using GPS [7]–[9]. To the best of our knowledge, the work in [1] is the first to demonstrate how a MAV could assist a ground robot in close collaboration in mapping a damaged building indoors. In their configuration, the ground-robot is equipped with a laser rangefinder and the flying robot with both a laser rangefinder and a RGBD sensor. The computed maps are aligned under the assumption that the ground robot does not move during the flight of the MAV. It is not mentioned whether the global map computation is executed onboard or in an offline stage and no processing times are reported. In our work, we investigate the relative localization, which is a precondition to the mapping task. Furthermore, we do not require that the ground-robot remains still while the flying robot is mapping and provide continuous relative position information in real-time.

Photorealistic modeling of urban scenes addresses a registration problem related to ours [10], [11]. Similar to our work, the one described in [12] addresses the computation of a 3D point-cloud from aerial video using dense motion stereo and the alignment with a ground-based map. Wendel [13] proposes to align a 3D reconstruction created by a MAV with a Digital Surface Model (DSM), where an initial alignment is provided by GPS information and a refined alignment is computed by evaluating the correlation between a height map computed from the reconstruction and the DSM. The time for alignment takes about 10 minutes, depending on the number of points and resolution. Our work advances the state of the art in two important aspects: (i) dense monocular maps are effectively used for MAV localisation and (ii) the integrated system can operate in real time, which is a desirable feature in most robotic perception tasks.

Registration methods based on image appearance require finding matches between the visual features in the aerial and ground views. Recently, advancements have been made in the field of wide-baseline image matching [6], [14], [15]. Many state-of-the-art approaches are grounded on the method described in [6] and aim at providing affine invariance by computing feature descriptors after a set of pre-defined warping transformations have been applied to the images to be matched. These methods implicitly rely on a piecewise planarity assumption, which is satisfied in many man made scenarios but does not hold in general. An example is provided in Figure 2. The aerial and ground views are shown from our validation dataset in case of indoor and outdoor scenarios. Figure 3 displays the results for a feature matching algorithm based on the work in [6] on the images corresponding to the ground and aerial views. Noticeably, the method managed to find few correct matchings on the planar box surfaces. The same method, applied to the views in the left column of Figure 2, returned no correct matches. The required processing time (about 6 seconds for feature extraction and 27 for matching) constitutes another important limitation to the application of robust approaches for visual feature matching to the problem of real time localisation.

To overcome these limitations, instead of relying on visual feature matching between the views from the MAV and the

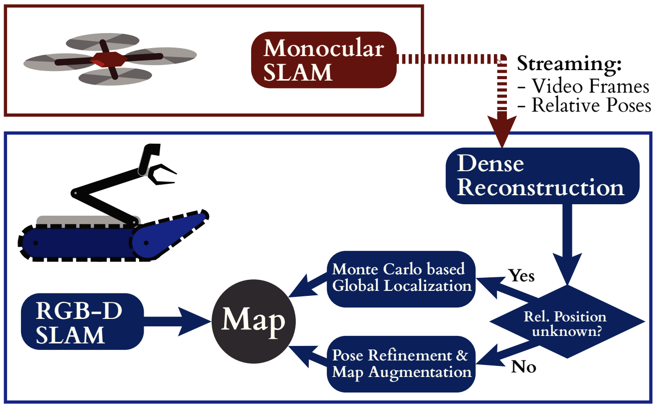


Fig. 4: Illustration of the localization and mapping pipeline. Each building block is described in detail in Sections IV to VII.

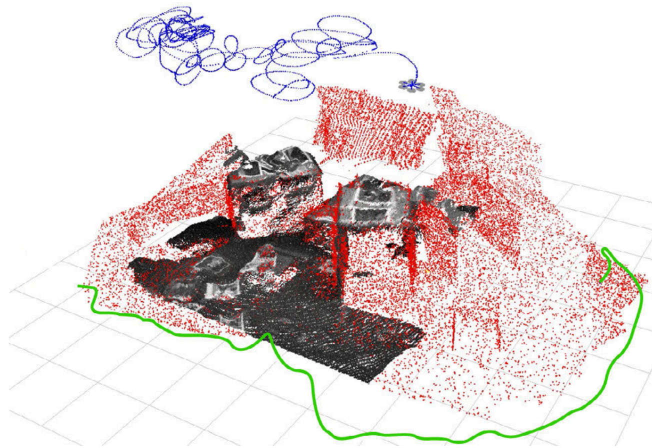


Fig. 5: Air-ground localization and map augmentation.The trajectories of the aerial and ground robots are displayed in blue and green respectively. The map computed by the ground robot (displayed in red) is augmented with the dense reconstruction from the aerial views (displayed in greyscale).

ground robot, our new approach exploits the 3D structure, which is computed by the MAV through monocular dense reconstruction and by the ground robot making use of its range sensor. This approach is novel and constitutes the actual contribution of this paper.

III. System Overview

Figure 4 provides an overview of the proposed system. The MAV is equipped with a single downward-looking camera and an IMU. A monocular SLAM algorithm runs onboard the MAV to estimate its egomotion. The absolute scale is recovered by a Kalman Filter [16]. The MAV streams the video to the ground robot together with relative-pose estimates for every frame.

On the ground robot, a set of subsequent frames received from the MAV are used to compute a *dense* map through leveraging all information in the monocular images—not only salient corner points. Real-time performance is achieved through a highly parallelized GPU implementation. The ground-robot is further equipped with a Kinect sensor to create a second—ground-based—3D map by means of an RGBD SLAM system.

For the alignment of the aerial and ground maps, we propose two solutions: If an *a priori* guess is available for the relative pose of the two robots, their maps are aligned using ICP. Otherwise, we propose a *Monte Carlo Localization* (MCL) based method to globally localize the MAV with respect to the ground robot. The MCL method relies on correlating the height-maps computed from the two vantage points.

The proposed pipeline requires an overlap between the aerial and ground maps and a 3D structure in the scene. In a completely flat environment, the algorithm does not converge. Hence, the proposed method is a strong complement to image feature based methods which fail to match in cluttered environments at such radically diffent viewpoints.

IV. SLAM on the MAV and the Ground Robot

The SLAM system on the flying robot implements the keyframe-based monocular *Visual Odometry* (VO) pipeline by Kneip et. al [17]. It is boosted in terms of robustness and efficiency by including incremental relative rotation priors obtained from the onboard IMU.

On the ground robot, an RGBD sensor is used to create the map. Our RGBD SLAM system is a modification of the monocular SLAM algorithm described above. Notably, we are able to speed-up triangulation using the depth provided by the sensor as a prior. Additionally, the depth measurements are used to initialize map-points in case of pure rotation of the camera.

Both SLAM systems could also be replaced with state-of-the-art algorithms such as [18]–[21].

V. Dense Monocular Reconstruction

In this section we describe a method to estimate the dense point cloud from the images collected by the MAV as it flies over an area of interest. Timestamped views and camera poses are streamed to the ground robot, where the computation can take advantage of the multi-cores architecture offered by the onboard GPU, an Nvidia Quadro K2000M in our experiments.

The solution we propose to estimate a dense point cloud from monocular views with known camera motion derives from *Multi View Stereo* methods [22] and is motivated by the following facts: i) assuming constant brightness, frames taken from close viewpoints allow high quality matching; ii) a large baseline among views enables a more reliable depth estimation and outlier rejection. Therefore, similarly to [3], we propose to compute depth maps from a large number of highly overlapping views, yielding a coarse, but very fast estimation. Filtering and regularization have been proposed to improve the accuracy of the depth maps computed from aggregation of the photometric error in stereo [3], [4]. Being computed from close views, these depth maps may still contain wrong estimations. For this reason, we chose to integrate several depth maps, which are computed as the MAV flies over the area of interest. This is due to the fact that—differently from those previous works mainly concerned with the recovery of visually appealing reconstructions—we are interested in accurate maps, useful for localization. Thus,

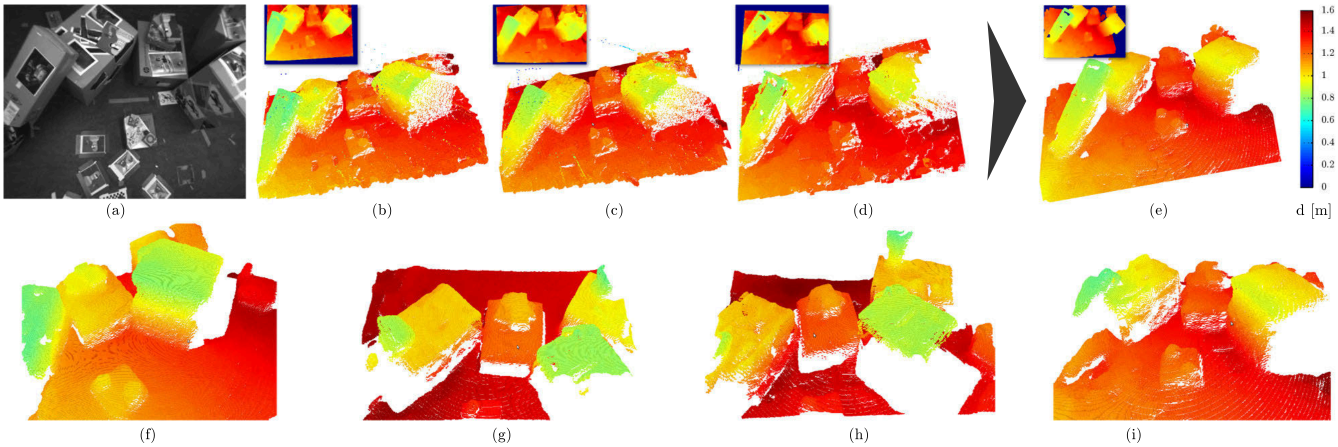


Fig. 6: Point clouds computed by monocular dense reconstruction for one indoor evaluation dataset. Depth maps are also shown in insets. In (a) the reference view is shown. Figures (b)-(d) correspond to different depth computations, fused into the depth map of Figure (e) through the algorithm presented in Section V. Figures (f)-(i) show several results from the fusion algorithm computed as the MAV flies over the experimental scenario. The color code refers to the distance d from the camera acquiring the reference view.

we aim at rejecting wrong estimations that would negatively affect the alignment performance. Further, the use of the recovered structure for the air-ground localization imposes severe constraints in computing time (cfr. Table I for average computing times). We chose to rely on the range-image fusion algorithm introduced in [23]. The algorithm is robust against wrong estimations; it is reported to be accurate and it is highly parallelizable, which makes it suitable for computation on a modern graphics card.

The depth maps are converted into distance fields $f_i : \Omega \to [-1, 1]$ defined on a voxel space $\Omega \subset \mathbb{R}^3$ specifying the volume of interest, and compressed into a histogram representation to reduce the memory footprint. At every voxel v, the values of f_i encodes the distance to the surface according to the i-th depth map and is approximated by evenly-spaced bin centers c_j.

Let $n(v, j)$ denote the histogram count of bin j at voxel v. The depth map fusion consists in estimating the distance field ϕ given the hypotheses represented by f_i and is computed by the following minimization of an energy functional:

$$\min_{\phi} \int_{\Omega} \left\{ |\nabla\phi| + \lambda \sum_{j} n(v,j)|\phi(v) - c_j| \right\} dv. \quad (1)$$

The data term $\sum_j n(v,j)|\phi(v) - c_j|$ approximates the distance of the solution from the distance fields, while the regularization term $|\nabla\phi|$ penalizes the surface area, removing errors due to outliers and approximating the surface in case of missing depth data. λ is a tunable parameter weighting the data term. The minimization in Equation 1 follows an iterative approach based on gradient descent.

The integrated surface is implicitly defined by the zero level set of the ϕ function and a point cloud is then computed through ray casting (see, for example, [24] Listing 3). Figure 6 depicts the process of fusing 3 dense structure estimations by the MAV (subfigures (b)-(d)) into one regularized structure (subfigure (e)). The algorithm effectively rejects erroneous estimations that are not supported by the majority of the depth maps. Different examples of dense reconstructions from the MAV views are reported in the second row of Figure 6. Once computed, the structure is made available for localization, as explained in the following sections.

VI. Global Localization

In this section, we describe a method to globally localize the MAV with respect to the ground robot based on 3D point-clouds computed from the two perspectives. Since the MAV operates in close range to the ground robot, we assume the global search region to be approximately 3m around the ground robot position.

The standard method to align two image-based maps is to find corresponding features (points, lines, edges, planes) either in the 2D images or the 3D pointcloud [25], [26]. However, we want to make no assumption on any regularity in the scene such as piecewise planarity. Additionally, both the aerial and ground based map can contain missing data and may not be fully overlapping.

We solve the problem through searching for the highest correlation between height-maps computed from the two pointclouds. In order to deal with local minima of the alignment, the procedure is extended with a Monte Carlo Localization method that verifies many hypotheses over the course of multiple pointclouds computed by the MAV. This extension is described thereafter.

A. Height-Map Alignment

In our setting, both the MAV and ground robot are equipped with an IMU. This provides the gravity vector, which can be used to project their maps to the ground plane (see Figure 7). This results in the height maps that we denote with H_a and H_g respectively. The height maps are defined on discrete 2D grids Ω_a and Ω_g with a resolution of 50 cells per meter. If multiple points project on to the same grid cell, the highest point is selected. Furthermore, we apply a morphological dilation operator on the height-maps of 3×3 cells in order to fill holes. Holes denote cells of the height-map with missing height data.

The best alignment of the two height maps in the predefined search region is found by searching for the relative pose

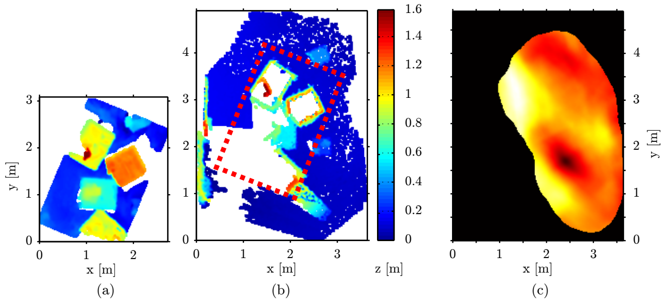


Fig. 7: The height maps of an aerial and a ground-based map are illustrated in Figures (a) and (b) respectively. The red dotted square in (b) shows the best alignment of the two height-maps. The costmap in (c) illustrates the ZMSSD cost for every possible relative position $\mathbf{u}$ of the two maps. A global minima is located at the dark spot.

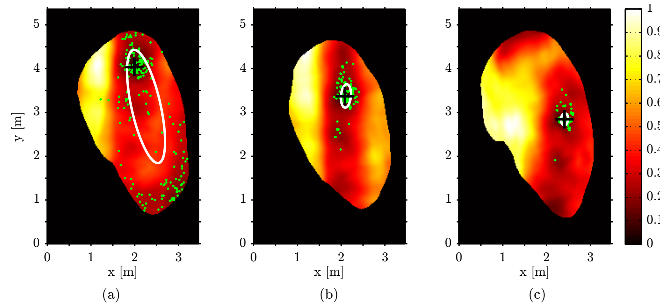


Fig. 8: Evolution of the particles in the Monte-Carlo–based global alignment. The costmap in the background is computed with Eq. (2) for all possible relative positions of three aerial maps. Dark values mean low ZMSSD. The white ellipse shows the covariance of the 200 green particles. The true position is marked with a black cross.

$\mathbf{u}$ with the minimum *Zero Mean Sum of Squared Differences* (ZMSSD) of the two maps:

$$C(\mathbf{u}) = \eta \sum_{\mathbf{x} \in \bar{\Omega}(\mathbf{u})} \left\{ [H_a(\mathbf{x}) - \hat{H}_a] - [H_g(\mathbf{x}+\mathbf{u}) - \hat{H}_g(\mathbf{u})] \right\}^2, \quad (2)$$

where $\hat{H}_g(\mathbf{u})$ and $\hat{H}_a$ are the mean of the height maps in the overlapping area at the relative position $\mathbf{u}$ and $\eta = 1/(2|\bar{\Omega}(\mathbf{u})|)$ is a normalization factor. Furthermore, $|\bar{\Omega}(\mathbf{u})| = |\Omega_a(\mathbf{u}) \cap \Omega_g|$ denotes the number of cells on which a height is defined for both, the translated aerial height-map $H_a(\mathbf{u})$ and the ground-based height-map H_g. The final relative position $\tilde{\mathbf{u}}$ corresponds to the minimum ZMSSD:

$$\tilde{\mathbf{u}} = \arg\min_{\mathbf{u}} \; C(\mathbf{u}) \quad (3)$$

The advantage of the ZMSSD cost is that the local normalization makes the alignment independent of the z location, whereas the final alignment in the vertical z axis can easily be recovered with $\Delta z = \hat{H}_g(\tilde{\mathbf{u}}) - \hat{H}_a$. However, the ZMSSD cost does not equalize the average relative heights between the two height-maps which is in contrast to the correlation cost which is applied in [13].

The search is done over $\mathbf{u} = [x, y] \in \mathbb{R}^2$ since in the experiments the magnetic north direction could be recovered from the IMU. Depending on the environment, this measurement can be less reliable which would require to add the heading direction to $\mathbf{u}$. This extension is straightforward but has the drawback that the computation time increases exponentially and there can be more local minima in the cost which can however be recovered with the filter described in the next section.

Furthermore, the search space is limited by a minimum overlap between the two height-maps $|\bar{\Omega}(\mathbf{u})| > \theta_{\text{overlap}}$. We set the threshold to 50% of the area of the aerial height-map $|\Omega_a|$. This is the reason for the curvy boarder in the costmap illustrated in Figure 7c.

B. Monte Carlo-based Alignment

Due to self-similarities in the environment, the costmap computed in the previous section may contain multiple local minima. We propose to apply Monte Carlo Localization with mixture proposal distribution [27, p. 262]. This allows us to track multiple hypotheses over the course of several height-maps computed from the MAV in order to identify the true relative position. We represent the belief of the relative position with a set $\mathcal{U}$ of M particles:

$$\mathcal{U} = \mathbf{u}^{[1]}, \mathbf{u}^{[2]}, \ldots, \mathbf{u}^{[M]}. \quad (4)$$

For the first height-map from the MAV the cost for each relative position is computed within the search region which results in the costmap of Figure 7c. M particles are then sampled from the costmap with probability

$$p(\mathbf{u}) \;\sim\; \exp\left\{ -\frac{C(\mathbf{u})}{\sigma} \right\}, \quad (5)$$

where σ depends on the resolution of the costmap and has been set to 0.08 in our experiments. This results in an initial distribution of the particles that spreads them among the local minima. When a new height-map is available from the MAV, the particles are propagated with the following motion model:

$$\mathbf{u}_t^{[n]} = \mathbf{u}_{t-1}^{[n]} + \Delta\mathbf{u} + \epsilon, \qquad \text{where } \epsilon \sim \mathcal{N}(0, \Sigma). \quad (6)$$

The relative motion $\Delta\mathbf{u}$ is provided by the SLAM on the MAV (Section IV).

Using the cost from Equation (2), the propagated particles are weighted and resampled. Hence, the full cost map of the search region is only required for the first aerial map to guarantee a good initial distribution of the particles. In each subsequent step, the cost is only computed at the M particle locations.

In the experiments we found that the particles converge to the true location after maximally three to four iterations (see Figure 11).

VII. Pose Refinement

Given an initial guess of the relative pose between the MAV and the ground robot, the relative pose can be refined through alignment of the respective pointclouds using ICP [28]. In order to assure convergence to the global minima, ICP needs to be initialized close to the solution; hence, the global alignment in the section above. Furthermore, the structure of the two pointclouds must be such that their relative movement is constrained (e.g., through both horizontal and vertical structures).

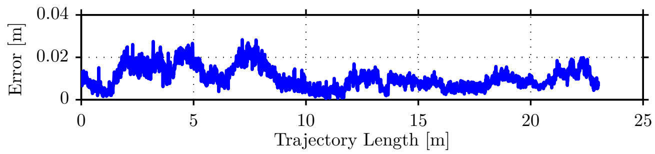


Fig. 9: Translation error of the Monocular SLAM on the MAV. The trajectory is 23.0 meters long and the RMS error is 1.2 cm. There is no visible drift because the trajectory contains many loops.

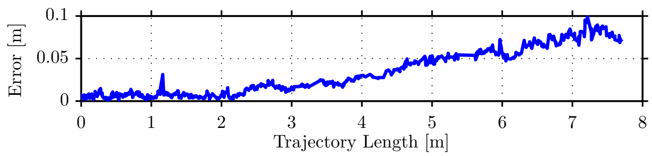


Fig. 10: Translation error of the RGBD SLAM on the ground robot. The trajectory is 7.7 meters and the RMS error is 3.8 cm.

We use the modular ICP algorithm *libpointmatcher* [21] that is provided as open-source software. To find the nearest-neighbour points, we apply a kd-tree which is provided by libpointmatcher. As an error metric, we use the point-to-plane distance.

In the experimental-results section, we demonstrate that the pointclouds computed from the dense reconstruction can be aligned with the ground-based map using ICP with an accuracy of 8 cm. Furthermore, we show that the alignment result from the previous section can be refined through ICP.

VIII. Experimental Results

We validated the proposed system on four datasets, three were collected indoors and one outdoors. The datasets consist of video and IMU recordings from both, the aerial and ground robot's point of view. The indoor environments were created out of cardboard boxes to resemble a disaster scenario (see Figure 2). Additionally, the ground-truth robot trajectories were recorded indoors with a motion-capture system. A video of the experimental results is available at `http://rpg.ifi.uzh.ch`.

A. SLAM Results

Figure 9 and 10 illustrate exemplary the translation error of the SLAM algorithms on the MAV and the RGBD ground robot respectively. Notice that the trajectory of the MAV does not drift. This is because the MAV flies several loops contrary to the ground robot. The Root-Mean-Square (RMS) error of the Monocular SLAM trajectory is 1.2 cm and for the ground robot 3.8 cm. Average timings of the algorithm are provided in Table I. The RGBD SLAM algorithm is slightly slower because on average more map-points were triangulated and tracked.

B. Dense Reconstruction

The map computed by the monocular SLAM of Section IV provides sparse information on the scene observed by the MAV and is conveniently used to determine the extent of the current volume of interest. The set of consecutive views

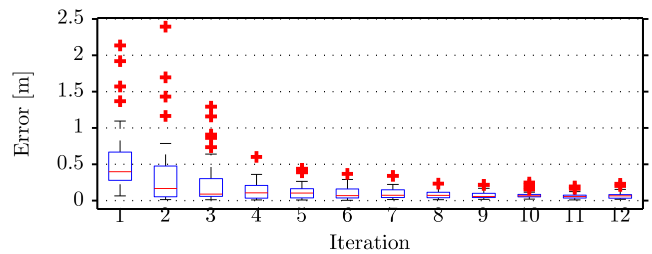


Fig. 11: Distribution of the translation error over 12 iterations of the Monte-Carlo–based localization illustrated with boxplots. The central mark on the box is the median, the edges of the box are the 25th and 75th percentiles. Results are from 41 experiments.

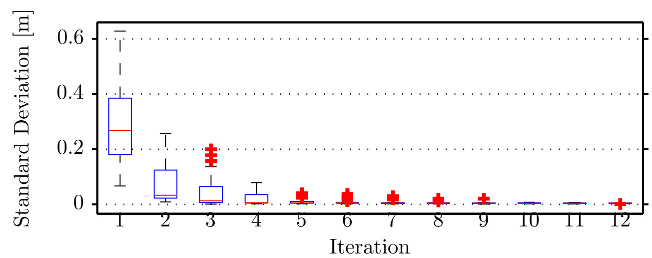


Fig. 12: Distribution of the standard deviation of the particles (see ellipses in Figure 8) over 12 iterations of the Monte-Carlo–based localization illustrated with boxplots. Results are from 41 experiments.

that are aggregated to form a depth map is simply characterised by the distance from the reference camera pose, and controlled by a tuneable threshold parameter set to 12 cm in our experiments. Similarly, a threshold on the distance from which the first depth map has been acquired characterises the set of depth maps to be fused. This distance was set to 70 cm for the experimental validation. Despite the basic view selection strategy controlling depth map creation and fusion, the proposed approach proved highly effective in computing dense and accurate data for the air-ground registration. The λ parameter was set to 0.26, while 8 iterations proved a good tradeoff between speed and accuracy for the minimisation in Equation 1.

C. Global Localization

The Monte-Carlo–based localization algorithm was tested on 41 sequences of 12 subsequent depth-maps from three different indoor environments. In Figure 11 the distribution of the localization error is reported for all 12 iterations. In 65% of the experiments, the distance between the mean of the particle distribution at the first iteration and the true position is less than 0.5 meters. Hence, the global minima of the costmap could attract most of the particles. After 4 iterations, the particle means of 89% of the 41 experiments have moved closer than 0.25m to the ground truth. At this range, the ICP algorithm can further refine the pose. Note that the accuracy of the alignment could be further improved by increasing the resolution of the height-maps at the cost of higher computation times. The processing time (Table I) for

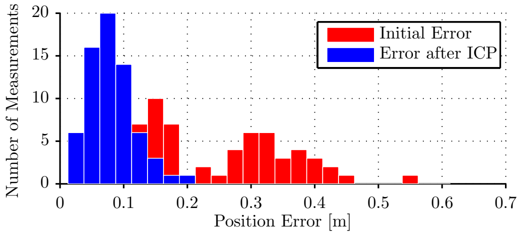


Fig. 13: Distribution of the translation error before and after ICP pose refinement. The data originates from 67 experiments in 3 different environments. The median error is 0.076 m.

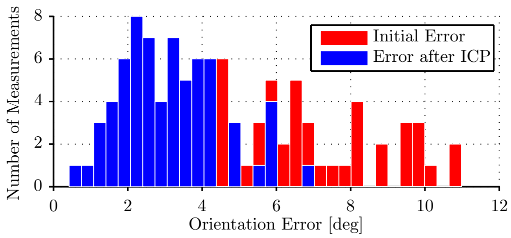
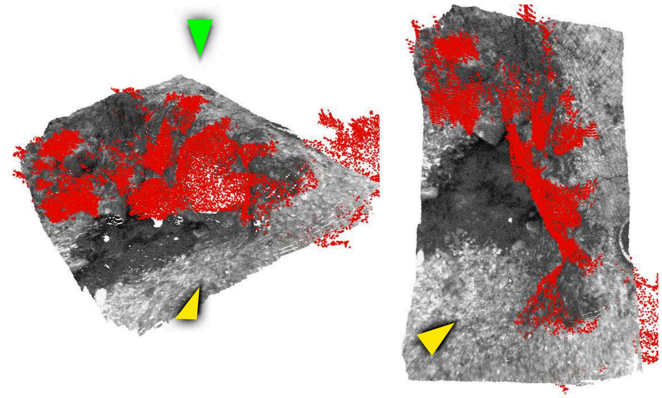


Fig. 14: Distribution of the angular error before and after ICP pose refinement. The provided angle derives from the angle-axis representation of the orientation error. The data originates from 67 experiments. The median error is 3.0 deg.

the first frame is approximately 9 seconds for the 4×5 meters search area on a single CPU. Furthermore, for every subsequent iteration, the ZMSSD cost must only be computed at the particle locations. We selected $M = 200$ particles which resulted in a processing time of approximately 38 ms on the CPU. Note that the processing time depends on the size of the depth-map, the search radius, and the number of particles. Furthermore, it was not necessary to inject new particles after the initial sampling. In order to detect whether the particles have converged, the covariance of the particle distribution can be considered, which is illustrated in Figure 12. One can observe, that with the convergence of the error also the variance decreases.

D. Pose Refinement

The pose refinement was tested with 67 depth-maps computed from the dense reconstruction in the three indoor environments. The translation and orientation errors before and after the alignement are reported in Figure 13 and 14. Since the monocular SLAM algorithm of the MAV is too accurate to illustrate the performance of the ICP algorithm, we artificially added Gaussian noise with $\sigma_{\text{ang}} = 3deg$ to the orientation, and $\sigma_{\text{trans}} = 0.2m$ to the position. The experiments show that the dense map computed by the monocular reconstruction is accurate and dense enough to succeed in the alignment with an accuracy of 8 cm and of 3 deg. There are two reasons why the error is not smaller: the ground map by the RGBD SLAM drifts (see Figure 10) or the error source could come from inaccuracies in the dense reconstruction. The ICP algorithm requires on average 0.5 seconds to align the maps. Note that all depth-maps contained 3D structures, which is a requirement for the ICP algorithm to converge to the global minima, as discussed



(a) (b)

Fig. 15: Results from the outdoor experiment. Figure (a) shows the aligned maps from the viewpoint of the ground robot (yellow triangle) and Figure (b) the same two maps from the aerial viewpoint (green triangle). The red pointcloud is computed from the ground robot and the dense greyscale pointcloud originates from the reconstruction of the aerial views. Refer to Figure 2 for two images from the dataset.

	Runtime [ms]
Egomotion Estimation:	
Monocular SLAM: Avg. time per frame	14
RGB-D SLAM: Avg. time per frame	15
Dense Reconstruction:	
Add frame to depth map (200 depth hypotheses)	5
Compute distance field from depth map (376 × 240 × 150 voxels)	21
Depth map fusion (8 iterations)	800
Ray casting	15
Global Localization:	
Full costmap computation (first depth-map)	9143
ZMSSD for $M = 200$ particles:	38
Pose Refinement:	
ICP alignment	462

TABLE I: Average runtimes of the algorithms in the pipeline. The timings were measured on an 8 core i7 laptop, 2.4 GHz. The used GPU is a Nvidia Quadro K2000M with 384 CUDA cores.

above. As soon as a map is available from the MAV, pose refinement is run on a dedicated thread. Given an acquisition rate of 30 frames per second, a new augmented map is available approximately every 300 frames.

Remarkably, the complete pipeline is capable of real-time performance on multi-core machines, and the timing for the complete execution is reported in Table I.

E. Outdoor Experiment

Figure 15 illustrates a result of the outdoor experiment. The same environment is also depicted in Figure 2. The dense reconstruction algorithm produced qualitatively highly accurate reconstructions due to the naturally very textured surface. The proposed pipeline succeeded in finding the correct alignment of the two maps. The accuracy cannot be reported as no groundtruth is available. Note that in this scenario state-of-the-art algorithms for wide-baseline visual feature maching normally fail as reported in Section II.

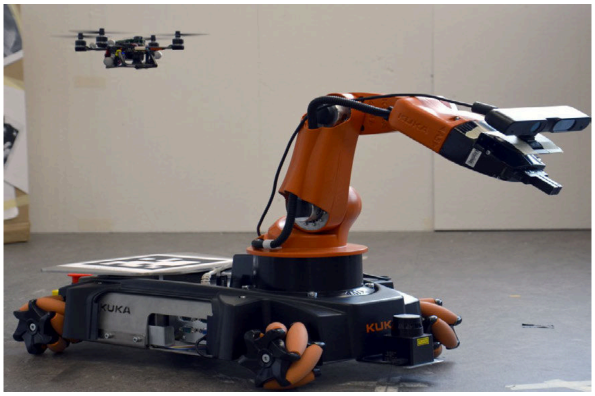


Fig. 16: The NanoQuad MAV, equipped with a down-looking camera and an onboard computer, is hovering above the Kuka YouBot ground robot, equipped with an RGBD camera.

IX. Conclusion

In this paper, we introduced a method to register the 3D structure computed by a MAV with that computed by a ground robot operating in close range. The MAV is equipped with a monocular camera while the ground robot relies on a range sensor. Building on the recent development of real-time, monocular dense mapping techniques, the proposed method allows the integration of structures computed from radically different viewpoints, i.e. from the aerial and the ground robot. Therefore, this paper contributes a novel approach to the fusion of visual maps with the ones computed from different depth-sensor modalities. Thereby, we prove how dense structure computation from monocular moving cameras is highly valuable in robot perception tasks. We demonstrated the effectiveness of the presented approach in augmenting the three-dimensional structure from the ground robot with an aerial dense map, in two different scenarios: three indoor, experimental setups, and one outdoor, where state-of-the-art alignment methods based on appearance normally fail.

References

[1] N. Michael, S. Shen, K. Mohta, Y. Mulgaonkar, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno, E. Takeuchi, and S. Tadokoro, "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," *Journal of Field Robotics*, vol. 29, no. 5, pp. 832–841, 2012.

[2] P. Rudol, M. Wzorek, G. Conte, and P. Doherty, "Micro unmanned aerial vehicle visual servoing for cooperative indoor exploration," in *IEEE Aerospace Conference*, 2008, pp. 1–10.

[3] R. A. Newcombe, S. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2320–2327.

[4] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," 2011, pp. 3017–3024.

[5] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments," *Journal of Field Robotics*, vol. 28, no. 6, pp. 854–874, 2011.

[6] J.-M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438–469, Apr. 2009.

[7] M. A. Hsieh, A. Cowley, J. F. Keller, L. Chaimowicz, B. Grocholsky, V. Kumar, C. J. Taylor, Y. Endo, R. C. Arkin, B. Jung, D. F. Wolf, G. S. Sukhatme, and D. C. MacKenzie, "Adaptive teams of autonomous aerial and ground robots for situational awareness," *Journal of Field Robotics*, vol. 24, no. 11-12, pp. 991–1014, Nov. 2007.

[8] T. Stentz, A. Kelly, H. Herman, P. Rander, O. Amidi, and R. Mandelbaum, "Integrated air/ground vehicle system for semi-autonomous off-road navigation," *Robotics Institute*, 2002.

[9] T. a. Vidal-Calleja, C. Berger, J. Solà, and S. Lacroix, "Large scale multiple robot visual mapping with heterogeneous landmarks in semi-structured terrain," *Robotics and Autonomous Systems*, vol. 59, no. 9, pp. 654–674, Sept. 2011.

[10] M. Ding, K. Lyngbaek, and A. Zakhor, "Automatic registration of aerial imagery with untextured 3d lidar models," *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2008.

[11] L. Liu and I. Stamos, "Multiview geometry for texture mapping 2d images onto 3d range data," *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2293–2300, 2006.

[12] W. Zhao, D. Nister, and S. Hsu, "Alignment of continuous video onto 3d point clouds," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 27, no. 8, pp. 1305 –1318, aug. 2005.

[13] A. Wendel, A. Irschara, and H. Bischof, "Automatic alignment of 3d reconstructions using a digital surface model," *Workshop IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29 –36, june 2011.

[14] M. Donoser and H. Bischof, "Efficient maximally stable extremal region (MSER) tracking," *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 553–560, 2006.

[15] C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys, "3d model matching with viewpoint-invariant patches (VIP)," *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2008.

[16] G. Nuetzi, S. Weiss, D. Scaramuzza, and R. Siegwart, "Fusion of IMU and vision for absolute scale estimation in monocular slam," *Journal of Intelligent and Robotic Systems*, vol. 61, p. 287299, 2011.

[17] L. Kneip, D. Scaramuzza, and R. Siegwart, "Robust Real-Time Visual Odometry with a Single Camera and an IMU," *Proc. of the British Machine Vision Conference (BMVC)*, 2011.

[18] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," *IEEE and ACM International Symposium on Mixed and Augmented Reality*, Nov. 2007.

[19] H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige, "Double Window Optimisation for Constant Time Visual SLAM," *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2011.

[20] N. Engelhard, F. Endres, J. Hess, J. Sturm, and W. Burgard, "Real-time 3d visual slam with a hand-held RGB-D camera," *Proc. RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum*, April 2011.

[21] F. Pomerleau, S. Magnenat, F. Colas, M. Liu, and R. Siegwart, "Tracking a depth camera: Parameter exploration for fast icp," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE Press, 2011, pp. 3824–3829.

[22] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, no. 8, pp. 1362–1376, 2010.

[23] C. Zach, "Fast and high quality fusion of depth maps," in *3DPVT*, 2008.

[24] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera," in *ACM Symposium on User Interface Software and Technology*, 2011, pp. 559–568.

[25] K. Leung, C. Clark, and J. Huissoon, "Localization in urban environments by matching ground level video images with an aerial image," *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 551 –556, may 2008.

[26] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D Registration," *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[27] S. Thrun, W. Burgard, , and D. Fox, *Probabilistic Robotics*. MIT Press, 2006.

[28] P. Besl and H. McKay, "A method for registration of 3-D shapes." *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 1992.