

Fast Localization and 3D Mapping using a RGB-D Sensor

Giuseppe Loianno, Vincenzo Lippiello, and Bruno Siciliano

Abstract—Low-cost range sensors represent an interesting class of sensors which are increasingly used for localization and mapping purposes in robotics. The combination of depth data and visual information can be employed to develop reliable algorithms for localization and environment mapping. A real-time approach combining a monocular visual odometry algorithm and range depth data is proposed in this paper. The scale factor problem is solved combining the depth data flow and the monocular image data. Moreover, a multiple resolution approach led by the distance of the sensor from surrounding obstacles is proposed for the depth data acquisition process. The visual egomotion estimation algorithm and the 3D map generation work in parallel improving the system real-time reliability. Experimental results show how the proposed integrated framework is able to localize in real-time the device in an unknown environment and to simultaneously generate an environment dense and colored map.

I. INTRODUCTION

Low-cost range sensors are an attractive alternative for expensive laser scanners or 3D cameras in research domains such as indoor navigation and mapping, surveillance and autonomous robotics. Consumer-grade range sensing technology gives the opportunity to choose between different devices available on the market like Microsoft Kinect sensor and ASUS Xtion sensor. They were primarily designed for natural interaction in a computer game environment (PrimeSense, 2010) using gesture recognition. However, the richness of the provided data and the low cost of the sensor have attracted many researchers from the field of mapping, 3D modeling and reconstruction. For example, the newest ASUS Xtion sensor presents a low weight with respect to the first generation of RGB-D cameras (around 70g without the external casing). It does not need external power other than the USB connection, and it is very compact, giving to this device some unique characteristics suitable, for example, for unmanned aerial vehicles applications [1].

A number of Simultaneous Localization and Mapping (SLAM) systems that make use of these sensors have already appeared in literature [2], [3], [4], [5], [6]. In particular, in [2] an RGB-D 3D mapping system utilizes a novel joint

optimization algorithm combining visual features and shape-based alignment. In [4] a direct 3D tracking approach is proposed such that an error is based directly on the intensity of pixels. Other algorithms are based on fusing depth maps to a coherent 3D model [5].

Algorithms based on 2D laser scanners [7], [8], 3D scanners [9], monocular cameras [10], [11], Inertial Measurement Unit (IMU) combined to a mono-camera system [12], [13], [14], [15] and stereo cameras [16], [17], [18], [19] have been widely used in the past for visual SLAM and visual odometry purposes. In [10] a monocular SLAM approach for small workspace is proposed, splitting the tracking and mapping process into two separate threads, whereas in [11] the application of a monocular technique with scale drift correction at loop closures is shown for large environments. Actually, it is worth underlining the promising results obtained with the application of monocular SLAM techniques to ground robots [3] and flying vehicles [6], as well as of monocular camera systems [12], [13], [20]. In [13] a kinect and on-board vehicle sensors are used to perform state estimation through Kalman filtering, while in [12], [20] a dynamic filter solves the scale factor estimation problem combining visual information with inertial sensor data. All these approaches show the feasibility of the adoption of these sensors also on a limited hardware/resource platform.

In the proposed work the new sensor ASUS Xtion is employed by coupling the monocular multi-map visual odometry algorithm proposed in [21] with depth data provided by the Infrared (IR) camera obtaining a real-time visual SLAM algorithm and a dense colored map. With the proposed solution the scale factor missing in [21] is estimated online such that the absolute map and position can be computed. The visual odometry algorithm works in a separate thread with respect to the depth generation map hence improving the reliability of the proposed algorithm on a multi-core platform. Moreover, the proposed system relies on the possibility to use both depth data and 3D features extracted from the monocular odometry algorithm improving the overall robustness of the map reconstruction algorithm. This is useful in case the device is used both in indoor and outdoor environment where the efficiency of the IR sensor can be locally compromised. Finally, to avoid memory rising up in large environment, a spatial multi-resolution approach is proposed to acquire point cloud data according to local environment distance. The framework has been developed under ROS [22] and is available to any user.

The rest of the paper is organized as follows: in Section II a review of the working principle of the RGB-D sensor

This work was supported by the ARCAS and SHERPA projects, funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under Grant ICT-287617 and ICT-600958, respectively. The authors are solely responsible for its content. It does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein.

The authors are with PRISMA Lab, Department of Electrical Engineering and Information Technology, University of Naples Federico II, via Claudio 21, 80125, Naples, Italy giuseppe.loianno@unina.it, vincenzo.lippiello@unina.it, bruno.siciliano@unina.it



Fig. 1. Asus Xtion Pro Live Sensor.

is provided showing how 3D points cloud are generated from depth data. In Section III it is shown how visual odometry is combined with depth data to estimate the scale factor. In Section V map generation is presented, whereas in Section VI experimental results are provided to show the effectiveness of the proposed approach. Section VII concludes the paper.

II. SENSOR CHARACTERISTICS

The ASUS Xtion sensor employed in this paper consists of an infrared laser emitter, an infrared camera and a RGB camera (see Fig. 1). Both the RGB and the depth images are provided with 640×480 pixels resolution at 30 Hz. The measurement of the depth data is achieved by a triangulation process [23]. The laser source emits a single ray which is split into multiple rays by a diffraction grating to create a constant pattern of speckles projected onto the scene. This pattern is captured by the infrared camera and is correlated against a reference pattern, which is obtained by capturing a plane at a known distance from the sensor. When a speckle is projected on an object whose distance to the sensor is smaller or larger than the one of the reference plane, the position of the speckle in the infrared image will be shifted in the direction of the baseline between the laser projector and the perspective center of the infrared camera. These shifts are measured for all speckles by an image correlation procedure, which yields a disparity image. For each pixel the distance to the sensor can then be retrieved from the corresponding disparity. In the following we consider a depth coordinate system which has origin at the perspective center of the infrared camera, with the Z axis orthogonal to the image plane towards the object, the X axis aligned to the direction of the baseline b between the infrared camera and the RGB camera. The Y axis is orthogonal to X and Z axis making a right-handed reference system.

Suppose that an object is on the reference plane at distance Z_f with respect to the sensor and the corresponding speckle is captured on the image plane of the infrared camera as shown in Fig. 2. If the object is shifted closer, far with respect to the sensor, the speckle on the image plane is displaced in the X direction, that corresponds to a disparity d in image space. The disparity d is strictly related to D , the displacement of the point k on the image, and to Z_k (depth), which denotes

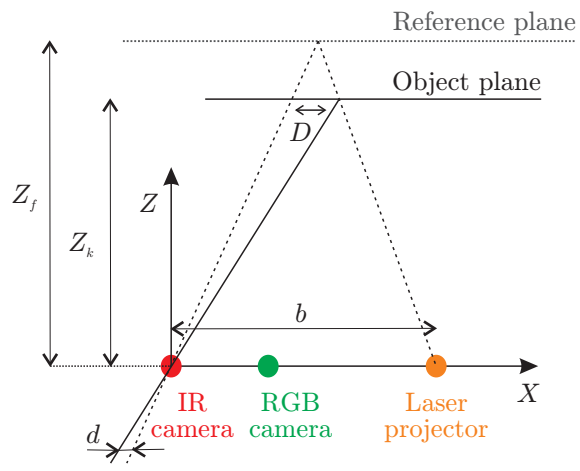


Fig. 2. Working Principle of IR Camera.

the distance of the point k as follows

$$\frac{D}{b} = \frac{Z_f - Z_k}{Z_f} \quad (1)$$

$$\frac{D}{f} = \frac{d}{Z_k}. \quad (2)$$

By plugging D extracted from (2) into (1), the following relation is obtained

$$Z_k = \frac{Z_f}{1 + \frac{Z_f}{fb}d}, \quad (3)$$

where b represents the base length between the IR projector and the infrared camera, while f is the focal length. All these parameters can be determined with a suitable sensor calibration procedure. The other two coordinates of the object 3D position can be determined by the classical perspective projection model as follows

$$X_k = -\frac{Z_k}{f}(x_k - x_c + \delta_x) \quad (4)$$

$$Y_k = -\frac{Z_k}{f}(y_k - y_c + \delta_y), \quad (5)$$

where x_c, y_c represent the coordinates of the principal point, δ_x, δ_y the lens distortion correction parameters, and x_k, y_k the corresponding normalized image coordinates.

The main problem to recover all the coordinates of the observed environment consists in the impossibility to stream the actual measured disparities, probably due to bandwidth limitation. In fact, the disparity values are normalized in the range 0–2047 and streamed as 11 bit integers. A zero sensor disparity does not correspond, for this reason, to an infinite distance. A normalized disparity should be considered in lieu of the one previously defined, and the sensor disparity is related to a normalized disparity by the following relation

$$d_n = \frac{1}{8}(d_o - d), \quad (6)$$

where d_n is a normalized disparity, and d_o is an offset value particular to a given sensor device. The factor $\frac{1}{8}$ appears

because the values of d are in $\frac{1}{8}$ pixel units. With a monocular camera calibration procedure of the infrared camera the focal length, the distortion parameters, and the lens center of the camera can be retrieved. Moreover, the parameters b and d_o can be measured with a least-squares fitting procedure by employing in (2) the estimates of the 3D position of the corners of a calibration chessboard, the corresponding measured projections of the corners in the infrared image, and the corresponding disparity values.

III. VISUAL EGOMOTION AND UNSCALED MAPPING

The visual SLAM algorithm PTAMM [21] is employed in our framework in order to localize the device within the environment with a single camera. The SLAM task is split into two parallel task, namely the tracking task and the mapping task, which are executed with two different threads.

The tracking task is responsible for the tracking of salient features in the camera image determining camera position. This is done with the following steps: first, a simple motion model is applied to predict the new pose of the camera. Then the stored map points are projected into the camera frame, and the corresponding features FAST corners [26] are searched (data association). The orientation and position of the camera is refined such that the total error between the observed point features and the projection of the map points into the actual frame is minimized.

On the other hand, the mapping task uses a subset of all camera images called key frames, selected through a heuristic criteria, to build a 3D point map of the environment. After adding a new key frame, a batch optimization is applied to refine both the map points and the key frame poses. The main purpose is the minimization of the total error between the back-projected map points and the corresponding measures in the key frames realizing the so-called *bundle adjustment*.

There are several important differences between the key frame SLAM algorithm considered here and the standard filter based approach [25]. The adopted algorithm does not use an EKF based state estimation and does not consider any uncertainties, both for the camera and the feature location, in a way to reduce the computational effort. However, considering the uncertainty of the state could ease the data association process. The lack of modeling uncertainties is compensated by the use of a large number of features and the global batch optimization. Therefore, despite the use of a fixed area for feature matches, the algorithm is still able to track efficiently the point features and to close loops. This makes the algorithm extremely fast and reliable and the map very accurate. As demonstrated in [24], the key frame SLAM outperforms the classical filter-based approach SLAM.

With respect to [10], users can now save the state of a map to disk, and then later reload it. Maps are saved in an open format using XML for the map data, and PNG images for the key frames. Only the salient data is saved, while some is regenerated on loading (such as FAST corners). A new class to handle thread synchronization issues is added to the Map class. The serializer has the complete control of a map, so other threads cannot corrupt the data during saving and

loading. All threads now have to register to each map and unregister when leaving. If a map is locked they will signal an acknowledgement and wait until the map is unlocked. The key advantage is that multiple maps can be saved in the background, with only a brief pause in operations in the current map during saving.

IV. SCALE FACTOR ESTIMATION

The adopted visual SLAM framework based on PTAMM can provide the translational motion of the sensor in the environment up to a scale factor, since a single camera is employed. The estimation of the scale factor parameter through the combination of depth data and visual data is proposed in this paper. We assume that the intrinsic and the extrinsic calibration parameters are available for both the IR and for the RGB camera. Let $\mathbf{p}_i^{IR}$ be the i -th point, expressed in the IR frame, which belongs to the point cloud

$$\mathbf{P}^{IR} = [\mathbf{p}_1^{IR}, \dots, \mathbf{p}_n^{IR}]^T, \quad (7)$$

which is generated as specified in Section II. The previous points cloud can be represented into the RGB frame as follows

$$\mathbf{p}_i^{RGB} = \mathbf{t}_{IR}^{RGB} + \mathbf{R}_{IR}^{RGB} \mathbf{p}_i^{IR}, \quad (8)$$

where $\mathbf{R}_{IR}^{RGB}$, $\mathbf{t}_{IR}^{RGB}$ are the rotation matrix and the translation vector of the IR camera with respect to the RGB camera, respectively, which are provided by the sensor calibration.

The depth values corresponding to each feature extracted within the visual framework have to be identified to evaluate the 3D point cloud. The adopted procedure is the same as described in Section II since the depth is actually expressed in the RGB frame. Let $\mathbf{c}_v$ be the center of mass of the points in the RGB visual framework, and $\mathbf{c}_d$ the corresponding center of mass for the depth points $\mathbf{p}_{v,i}$, with $i = 1, \dots, n$. The ratio between the Euclidean distance of each 3D point, generated in the visual framework, from the FAST corners [26], with respect to its center of mass

$$\Delta p_i^{RGB} = \|\mathbf{p}_i^{RGB} - \mathbf{c}_v\|, \quad (9)$$

and the distance computed for the corresponding depth points

$$\Delta p_{v,i} = \|\mathbf{p}_{v,i} - \mathbf{c}_d\|, \quad (10)$$

yields a set of scale factors

$$\mathbf{s} = [s_1, \dots, s_n]^T, \quad \text{where} \quad s_i = \frac{\Delta p_i^{RGB}}{\Delta p_{v,i}}. \quad (11)$$

The current estimation of the scale factor corresponds to the mean value of $\mathbf{s}$. A statistical procedure is setting up deleting the values out of the statistics of a normal distribution to improve the achieved measure. A synthetic representation of the estimation process is given in Algorithm 1. Notice that the combination of the depth and visual data to recover global consistency is strictly required only for the first frame, as well as the values s_i could be computed only for the extracted corners. The proposed solution is computational inexpensive and allows the evaluation of the environment map directly from the cloud points given by the IR/depth image.

Algorithm 1 Scale.Factor.Estimation($RGB_{image}, Depth_{image}$)

```

extract_image_features.set( $RGB_{image}$ )  $\rightarrow$   $\mathbf{F} = [\mathbf{f}_1, \dots, \mathbf{f}_n]^T$ ;
compute_3D_points( $RGB_{image}, \mathbf{f}$ )  $\rightarrow$   $\mathbf{p}_v$ ;
compute_center_of_mass( $\mathbf{p}_v$ )  $\rightarrow$   $\mathbf{c}_v$ ;
compute_depth_points( $Depth_{image}$ )  $\rightarrow$   $\mathbf{p}_i^{IR}$ ;
roto.translate( $Depth_{image}$ )  $\rightarrow$   $\mathbf{p}_i^{RGB} = \mathbf{t}_{IR}^{RGB} + \mathbf{R}_{IR}^{RGB} \mathbf{p}_i^{IR}$ ;
compute_center_of_mass( $\mathbf{p}^{RGB}$ )  $\rightarrow$   $\mathbf{c}_d$ ;
for  $\forall \mathbf{f}_i$  do
     $\mathbf{f}_i \rightarrow depth_i$ ;
     $\Delta p_i^{RGB} = \|\mathbf{p}_i^{RGB} - \mathbf{c}_v\|$ ;
     $\Delta p_{v,i} = \|\mathbf{p}_{v,i} - \mathbf{c}_d\|$ ;
     $s_i = \frac{\Delta p_i^{RGB}}{\Delta p_{v,i}}$ ;
end for
 $\mathbf{s} = [s_1, \dots, s_n]^T$ ;
is_deleted = true;
while is_deleted do
    is_deleted = false;
     $s_f = \text{mean}(\mathbf{s})$ ;
     $\sigma = \text{std}(\mathbf{s})$ ;
    for  $\forall i$  do
        if  $\text{abs}(s_i - s_f) \geq 2\sigma$  then
            delete  $s_i$ ;
            is_deleted = true;
        end if
    end for
end while
return  $s_f$ 

```

V. 3D MAPPING AND RECONSTRUCTION

The environment map generation is computed in parallel with respect to visual odometry evaluation. The map is generated through the depth values using the monocular visual odometry framework. After the algorithm initialization, when the scale factor estimation is performed, the absolute position and orientation are computed. These values are then associated to the point clouds generated from the IR sensor, which is composed of around 300.000 points if a depth image of resolution 640×480 pixels is considered. The result is a colored map given by combination of the RGB image pixels colors with the depth values.

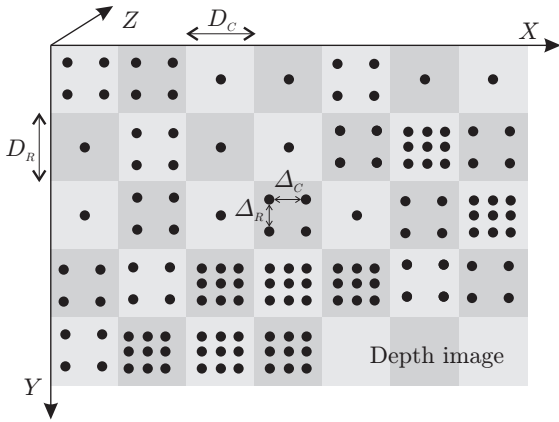


Fig. 3. Depth image multi-resolution sampling.

To avoid an excessive memory allocation, which can affect computation performances, without reducing the spatial accuracy of the environment map, a new multi-resolution approach is proposed for point clouds sampling. For each IR

image, the corresponding point clouds are divided in different groups through a virtual image grid, which is composed of rectangles of size $D_R \times D_C$ pixels (see Fig. 3). In particular, for each rectangle of the grid, a spatial sampling with a step $\Delta_R \times \Delta_C$ is chosen accordingly to the following laws

$$\Delta_R = \Delta_{R,min} + \frac{\Delta_{R,max} - \Delta_{R,min}}{Z_{max} - Z_{min}} (Z_{max} - Z_C) \quad (12)$$

$$\Delta_C = \Delta_{C,min} + \frac{\Delta_{C,max} - \Delta_{C,min}}{Z_{max} - Z_{min}} (Z_{max} - Z_C), \quad (13)$$

where Z_C is the third component of the point centered into the current rectangle or the average distance between the rectangle and the environment expressed into the IR frame. The terms $\Delta_{R,min}$, $\Delta_{R,max}$, $\Delta_{C,min}$, and $\Delta_{C,max}$ represent the minimum and maximum values for the spatial sampling distance, respectively. According to the computed value, a certain number of points is added to the global map. For each rectangle the number of points, added to the global map, varies as shown with a linear law between a maximum distance Z_{max} and the minimum distance Z_{min} , which the sensor is able to detect, according to the distance from the environment Z_C in the current cell.

Moreover, a sampling time is adopted depending on the sensor linear and angular velocity. In details, the map publishing thread takes care of the map streaming and pose visualization in real-time, while the map data storage is updated according to the following time laws

$$F = F_{max} - (F_{max} - F_{min}) \cdot \max\left(\frac{v}{v_{max}}, \frac{\omega}{\omega_{max}}\right) \quad (14)$$

where F is the map update frequency, F_{max} , F_{min} are the maximum and minimum publication updating frequencies rate, v and ω are the mean linear and angular velocity norms, respectively, performed on the last k time instants as follows

$$v = \frac{1}{k} \sum_{i=1}^k v_i \quad \omega = \frac{1}{k} \sum_{i=1}^k \omega_i. \quad (15)$$

The quantity v_i (ω_i) is the linear (angular) velocity norm at time instant i computed as follows

$$v_i = \frac{1}{\Delta T} (p_i - p_{i-1}) \quad (16)$$

$$\omega_i = \frac{1}{\Delta T} \theta_{i-1}^i, \quad (17)$$

where ΔT is the time interval between two consecutive measurements, p_i is the position norm at time instant i , while θ is obtained from the axis-angle orientation representation of two consecutive time rotations

$$(\mathbf{R}_i \mathbf{R}_{i-1}^T) \rightarrow (\mathbf{1}, \theta_{i-1}^i). \quad (18)$$

The main problem for this kind of sensors is their efficiency in outdoor environments where depth generation is compromised due to outside light intensity. For this reason the presented framework gives the opportunity to use, instead of IR camera points, 3D points in the fixed frame generated by the visual framework. The main difference with respect to [21] is that all map points are scaled by the estimated scale factor generating an absolute environment map.

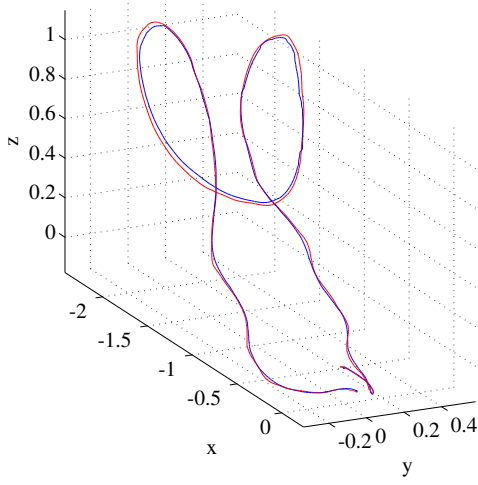


Fig. 5. Path trajectory of the sensor (in red) and the corresponding ground-truth (in blue) provided by the Optitrack motion capture system.

VI. EXPERIMENTAL RESULTS

The OpenNI driver [27] has been employed for the sensor interfacing, which provides the capability of choosing between different configurations in terms of image resolution and updating frequency. For the considered experimental case study, the RGB and depth data are streamed with a frequency of 30 Hz and resolution of 640×480 pixels. The proposed framework has been encapsulated in a ROS [22] node and the time synchronization between RGB image and depth is realized via the ROS message synchronization mechanism. The map can be published up to 20 Hz ($F_{max} = 20$ Hz and $F_{min} = 10$ Hz), with $\Delta_{R,max} = 10$, $\Delta_{R,min} = 5$, $\Delta_{C,max} = 20$, $\Delta_{C,min} = 10$.

To show the effectiveness of the proposed approach the PRISMA Lab has been reconstructed. Figure 4 shows the original room (on the top) and the corresponding dense colored map (on the bottom), while the sensor trajectory is depicted with a blue line inside the map environment.

Moreover, to evaluate the effectiveness of the proposed scale factor estimation method, the sensor has been moved along a trajectory of about 8 m long, which is shown in Fig. 5. In order to provide a ground truth for the proposed egomotion estimation algorithm, an OptiTrack motion capture system [28] composed of ten S250e cameras has been employed to track the sensor during its motion at 250 Hz.

The time history of the norm of the motion estimation error with respect to the ground truth shown in Fig. 6 highlights a 15 cm peak over a mean of about 3.8 cm, which confirms the effectiveness of the proposed approach. In fact, the sensor accuracy for the depth measurement at 4 m of distance is declared in 3 cm, and thus the performance of the proposed approach is in line with the sensor intrinsic performance. The positional norm error is decreasing after 30 seconds since the camera is back to previous mapped positions, where keyframes have already

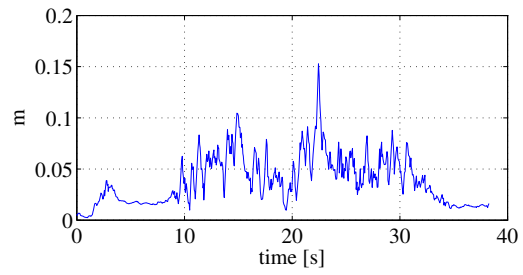


Fig. 6. Time history of the positional norm error for the path of Fig. 5.

been instantiated. The results can be seen in the video at http://wpape.unina.it/lippiell/videos/icar2013_video.mp4.

VII. CONCLUSION

In this paper a real-time method for pose estimation and mapping has been presented. A new scale factor estimation method using a combination of RGB monocular visual odometry and depth map has been proposed. The map is generated according to a multi-resolution sampling approach avoiding an unnecessary memory allocation in case of large environment. By using a multi-thread programming approach, the algorithm allows a fast localization and colored dense map reconstruction of the environment suitable for real-time applications.

REFERENCES

- [1] L. Marconi, L. Basile, G. Caprari, R. Carloni, P. Chiacchio, C. Huerzeler, V. Lippiello, R. Naldi, N. Janosch, B. Siciliano, S. Stramigioli, E. Zwickler, "Aerial Service Robotics: The AIRobots Perspective", *2nd Int. Conf. on Applied Robotics for the Power Industry*, Zurich, Switzerland, 2012.
- [2] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments", *International Symposium on Experimental Robotics (ISER)*, New Delhi, India, 2010.
- [3] N. Engelhard, F. Endres, J. Hess, J. Sturm, and W. Burgard, "Realtime 3D visual SLAM with a hand-held RGB-D camera", *Workshop on 3D Perception in Robotics at the European Robotics Forum*, Västerås, Sweden, 2011.
- [4] C. Audras, A. Comport, M. Meilland, and P. Rives, "Real-time dense appearance-based SLAM for RGB-D", *Australasian Conference on Robotics and Automation*, Melbourne, Australia, 2011.
- [5] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking", *International Symposium on Mixed and Augmented Reality (ISMAR)*, Basel, Switzerland, 2011.
- [6] S. Sean, N. Michael, and V. Kumar, "3D Indoor Exploration with a Computationally Constrained MAV", *IEEE Int. Conf. on Robotics and Automation*, St. Paul, USA, 2012.
- [7] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem", *National Conference on Artificial Intelligence (AAAI)*, USA, 2002.
- [8] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters", *IEEE Transactions on Robotics (T-RO)*, vol. 23, pp. 34–46, 2007.
- [9] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann, "6D SLAM-3D mapping outdoor environments: Research articles", *Journal of Field Robotics*, vol. 24, pp. 699–722, August 2007.
- [10] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces", *International Symposium on Mixed and Augmented Reality (ISMAR)*, Nara, Japan, 2007.

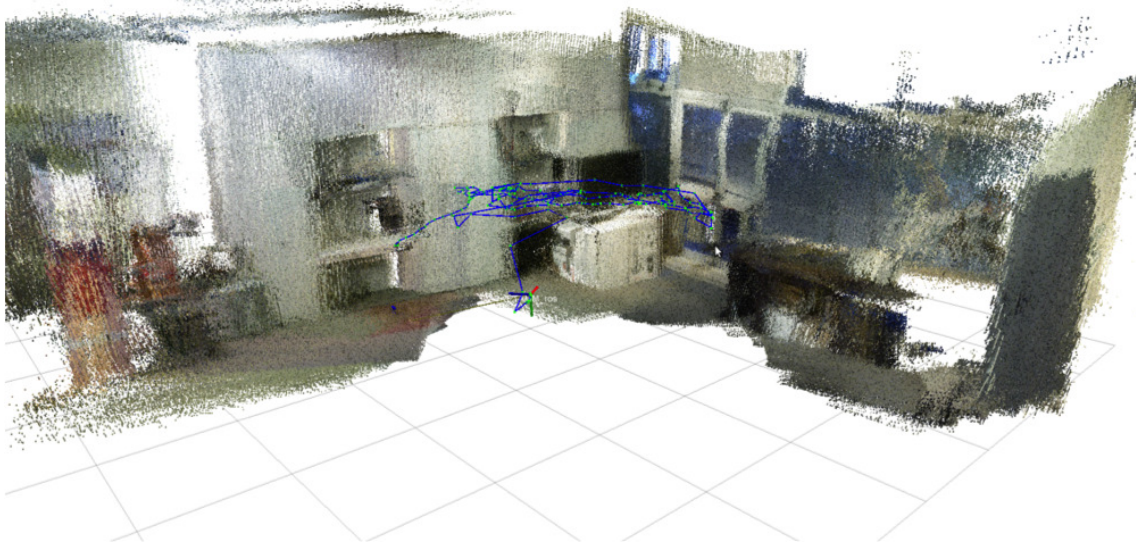


Fig. 4. Dense colored map reconstruction: on the top the real environment; on the bottom the achieved map. The blue line indicates the sensor motion within the environment as measured by the visual egomotion estimation algorithm with the proposed scale factor computation.

- [11] H. Strasdat, J. Montiel, and A. J. Davison, "Scale-drift Aware Large Scale Monocular SLAM", *Robotics Science and Systems (RSS)*, Zaragoza, Spain, 2010.
- [12] M. Achtelik, M. Achtelik, S. Weiss, and R. Siegwart, "Onboard IMU and Monocular Vision Based Control for MAVs in Unknown In- and Outdoor Environments", *IEEE Int. Conf. on Robotics and Automation*, Shanghai, China, 2011.
- [13] S. Sean, N. Michael, and V. Kumar, "Autonomous Multi-Floor Indoor Navigation with a Computationally Constrained MAV", *IEEE Int. Conf. on Robotics and Automation*, Shanghai, China, 2011.
- [14] V. Lippiello, G. Loianno, and B. Siciliano, "MAV Indoor Navigation Based on a Closed-Form Solution for Absolute Scale Velocity Estimation Using Optical Flow and Inertial Data", 50th IEEE Conference on Decision and Control and European Control Conference, Orlando, FL, USA, 2011.
- [15] V. Lippiello, R. Mebarki, "Closed-Form Solution for Absolute Scale Velocity Estimation Using Visual and Inertial Data with a Sliding Least-Squares Estimation", 21st Mediterranean Conference on Control and Automation, Crete, Greece, 2013.
- [16] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3d Reconstruction in Real-time", *IEEE Intelligent Vehicles Symposium*, Baden, Germany, 2011.
- [17] R. Voigt, J. Nikolic, C. Huerzeler, S. Weiss, L. Kneip, and R. Siegwart, "Robust Embedded Egomotion Estimation", *IEEE Int. Conf. on Intelligent Robots and Systems*, San Francisco, USA, 2011.
- [18] F. Donnarumma, V. Lippiello, M. Saveriano, "Fast Incremental Clustering and Representation of a 3D Point Cloud Sequence with Planar Regions", 2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Vilamoura, Portugal, 2012.
- [19] V. Lippiello, B. Siciliano, "Wall Inspection Control of a VTOL Unmanned Aerial Vehicle Based on a Stereo Optical Flow", 2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Vilamoura, Portugal, 2012.
- [20] L. Doitsidis, A. Renzaglia, S. Weiss, E. Kosmatopoulos, D. Scaramuzza, and R. Siegwart, "3D Surveillance Coverage Using Maps Extracted by a Monocular SLAM Algorithm", *IEEE Int. Conf. on Intelligent Robots and Systems*, San Francisco, USA, 2011.
- [21] R.O. Castle, G. Klein, and D. Murray, "Video-rate Localization in Multiple Maps for Wearable Augmented Reality", *IEEE International Symposium on Wearable Computers*, Pittsburgh, USA, 2008.
- [22] ROS, "Robotics Operating System", www.ros.org.
- [23] B. Freedman, A. Shpunt, M. Machline, and Y. Arieli, "Depth mapping using projected patterns", Prime Sense Ltd, USA, 2010.
- [24] H. Strasdat, J. Montiel, and A. J. Davison, "Realtime monocular SLAM: Why filter?", *IEEE Int. Conf. on Robotics and Automation*, Anchorage, USA, 2010.
- [25] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.29 (6), pp.1052–1067.
- [26] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking", *IEEE Int. Conf. on Computer Vision*, Beijing, China, 2005.
- [27] OpenNI, www.openni.org.
- [28] Optitrack, "Optical Motion Capture System and Tracking Software" <http://www.naturalpoint.com/optitrack/>.
- [29] D. Scaramuzza and F. Fraundorfer, "Visual Odometry Part I: The First 30 Years and Fundamentals", *IEEE Robotics and Automation Magazine*, vol. 18, pp. 80–92, December 2011.
- [30] F. Fraundorfer and D. Scaramuzza, "Visual Odometry Part II: Matching, Robustness, Optimization and Applications", *IEEE Robotics and Automation Magazine*, vol.19, pp.78–90, June 2012.