

Real-Time Depth and Inertial Fusion for Local SLAM on Dynamic Legged Robots

Marco Camurri¹, Stephane Bazeille¹, Darwin G. Caldwell¹ and Claudio Semini¹

firstname.lastname at iit.it

Abstract—We present a real-time SLAM system that combines an improved version of the Iterative Closest Point (ICP) and inertial dead reckoning to localize our dynamic quadrupedal machine in a local map. Despite the strong and fast motions induced by our 80 kg hydraulic legged robot, the SLAM system is robust enough to keep the position error below 5% within the local map that surrounds the robot. The 3D map of the terrain, computed at the camera frame rate is suitable for vision based planned locomotion. The inertial measurements are used before and after the ICP registration, to provide a good initial guess, to correct the output and to detect registration failures which can potentially corrupt the map. The performance in terms of time and accuracy are also doubled by preprocessing the point clouds with a background subtraction prior to performing the ICP alignment. Our local mapping approach, in spite of having a global frame of reference fixed onto the ground, aligns the current map to the body frame, and allows us to push the drift away from the most recent camera scan. The system has been tested on our robot by performing a trot around obstacles and validated against a motion capture system.

I. INTRODUCTION

The problem of Simultaneous Localization and Mapping (SLAM) has a long history: it has been theoretically defined and solved long ago [1][2]; but the many issues related to noise, synchronization, latency and rounding errors drive the research to find new methods and techniques to increase the robustness against these complications.

In the context of legged robots, even more problems arise, since some of the simplified hypotheses used in wheeled robots are not applicable, like the quasi-planar assumption for the ground which limits movements on the z -axis and roll/pitch motions. Furthermore, legged locomotion involves dealing with static and dynamic stability, and impacts that affects image sensing.

At the *Istituto Italiano di Tecnologia*, the fully torque-controlled Hydraulic Quadruped robot (HyQ) has been designed to perform agile and highly dynamic locomotion on difficult terrains [3]. It is a versatile robot that weighs 80 kg, is 1 m long and 1 m tall (Fig. 1, *Left*). The robot's legs have three degrees of freedom each, two joints in the sagittal plane (hip and knee flexion/extension) and one joint for hip abduction/adduction. HyQ is equipped with a 3DM-GX3[®]-25 IMU, rigidly attached to the robot base, a Bumblebee2 stereo camera and an Xtion depth sensor. The two image sensors are mounted on a Pan-and-Tilt Unit (PTU) which provides two degrees of freedom for active scanning purposes (Fig.

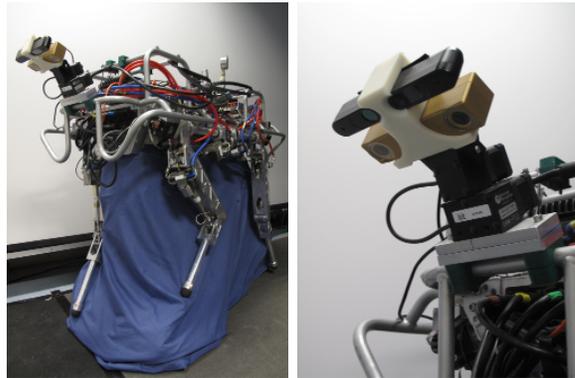


Fig. 1. The quadruped robot HyQ and its vision system. *Left*: the whole robot. *Right*: detailed view of the robot's active head consisting of an RGB-D camera and a stereo camera mounted on a pan and tilt unit.

1, *Right*). The results presented in Section VI are referred to the RGB-D sensor only, while the stereo is used for the calibration setup described in Section III-A. Future work will involve the use of the stereo camera for outdoor experiments, where RGB-D sensors becomes unusable due to sunlight.

During the past few years, we have shown several capabilities of this mechanical platform, among these are crawling, walking, trotting and jumping while the robot was teleoperated. Using perception sensors, we recently demonstrated some planned locomotion in prebuilt maps [4] and some preliminary results on onboard vision based locomotion [5].

The development of an online localization and mapping system is crucial to perform these tasks autonomously. Hence, we address the SLAM problem in a task oriented manner: since we are not interested in exploration, we decided to localize the robot within a map, whose size is suitable for short-term planning (*i.e.*, 4-5 step in advance). A particular attention has been given to the output frequency and the processing power requirement of the SLAM framework as we aim for onboard autonomous behaviors.

Contributions: This paper presents the development of a SLAM system for our legged robot that fuses visual 3D data and IMU data. The main contributions for this work are: (a) the development of an onboard real-time localization method that combines ICP-based registration enhanced by background subtraction of the associated depth image and inertial signals (b) the proposal of an alternative approach for 3D mapping on dynamic quadruped robots, which leverages

¹Dynamic Legged Systems Lab, Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova, Italy.

the robocentric nature of short-term planned navigation to push the drift error away from the operative area (c) a review on the main issues that affect SLAM on complex machines, which involved the development of a calibration technique based on fiducial markers and motion capture (d) the validation of the aforementioned methods and techniques on a real application with our dynamic legged robot.

Contents: The remainder of this paper is organized as follows: Section II describes the relevant works on fusion of depth/inertial sensors and on perception with legged robots. Section III lists the non-idealities that occur when performing mapping on a legged machine, and how we addressed them. Section IV describes our ICP-based registration method fused with inertial data for localization. Section V describes our proposed local mapping approach. Section VI shows our experimental setup and comments the results obtained with our hydraulic robot. Finally, Section VII summarizes the work presented herein and lists its future developments.

II. RELATED WORK

Many groups have recently suggested methods for fusing the output of depth and inertial sensors for SLAM, state estimation or scene reconstruction. Usually registration-based methods are used for dense scene reconstruction, while state estimation or SLAM methods generally rely on the Extended Kalman filter.

Nießner *et al.* [6] combined inertial measurements and depth sensor outputs for dense scene reconstruction. Their framework includes a method for detecting an ICP failure and switching to dead reckoning by IMU integration, to avoid bad reconstructions.

Schmid and Hirschmüller [7] presented an integrated device to estimate ego-motion of a camera pair using the depth from Semi-Global Matching and IMU data. The device includes also an FPGA to deal with the problem of synchronization (see Section III-B).

Qayyum and Kim [8] fused inertial and depth sensor with a modular EKF framework. They addressed the practical development of an Inertial-Kinect fused SLAM that works outdoor. They focused on handling the 3D to 2D degeneration in structured light sensing, called the depth dropout problem.

Bethencourt and Jaulin [9] introduced recently a new concept for point cloud registration based on an interval analysis method. They are not focusing on SLAM but achieved consistent Kinect point cloud matching using also the IMU data.

With quadruped robot, few groups have also presented some localization/state estimation solutions. For example Kolter *et al.* [10] use a stereo camera with a simple ICP-based technique for offboard point cloud registration to incrementally build a map. Then, they use a texture synthesis algorithm to fill occluded areas in order to perform motion planning with *LittleDog*.

Stelzer *et al.* [11] developed a complete visual navigation framework for their hexapod robot. The algorithm used stereo images from which depth images are computed. Pose estimates are obtained by fusing inertial data with relative

leg odometry and visual odometry measurements using an indirect information filter.

In a similar way, [12] fused the information from stereo vision, leg odometry, and IMU in order to obtain accurate state estimation of *BigDog*. On the above mentioned system they also developed a registration pipeline and a 2D cost map framework tailored for navigation.

Finally [13] proposed an elevation mapping method from a robot-centric perspective. The localization is performed using their state estimation framework based on kinematics and IMU and the mapping is obtained by fusing height maps from a Kinect.

Our work mainly differs from the ones presented here because it operates on a 3D domain, as typical scene reconstructions applications, but it is constrained to real-time work on limited onboard hardware, an operative condition typical of SLAM, where most SLAM methods use 2D representations. To achieve this, we developed a selection method to reduce drastically the input for ICP and keep the real-time constraint. Furthermore, all the other methods except [13] operate in a global reference frame.

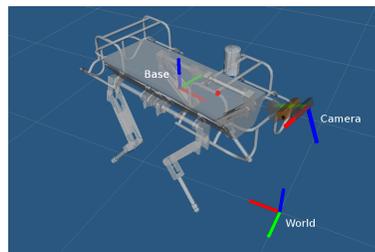


Fig. 2. The different reference frames used in this paper. The camera frame, the base frame and the world frame defined as the base frame position and orientation at $t = 0$.

III. NON-IDEALITIES

A. Robot transform calibration

Many of the setups described in the related work include an image sensor with an IMU rigidly attached to it. This makes the data fusion between the two sensors easy, since the displacement between their frames of reference is known or small enough to be neglected.

As our final goal is to perform SLAM for navigation, the target of state estimation is the robot base, and not the image sensor. Thus, a transform between the optical center frame O_c , and the robot base frame O_b is needed (see Fig. 2); but retrieving the transform ${}_bT_c$ is not straightforward, because of:

- 1) mechanical tolerances between assembly parts of the robot;
- 2) mechanical tolerances and distortion of the depth sensor (see Section III-C);
- 3) non-rigidity of the materials, which produce undesired motions when shocks occur;
- 4) two additional degrees of freedom, provided by the Pan-and-Tilt Unit, which extend the chain of transformations

to the robot base and thus propagate their respective errors.

While issue 3) is practically impossible to be eliminated, issues 1) and 2) can be solved through a static calibration, and issue 4) can be partially solved by storing a static calibration for each angle combination of the PTU.

Note that attaching a second IMU to the camera and performing sensor fusion would still need ${}_bT_c$ to perform SLAM.

To retrieve a valid transform ${}_bT_c$, we developed a calibration method that involves the use of a motion capture system and the augmented reality library ArUco [14].

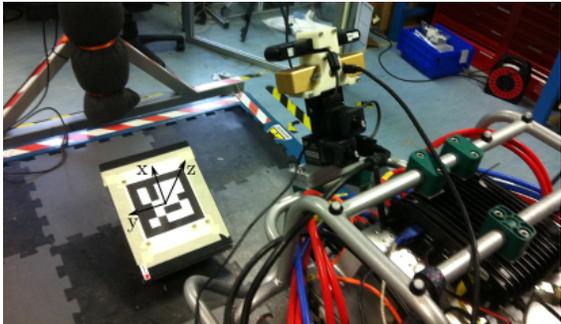


Fig. 3. Our calibration setup: a fiducial marker is surrounded by motion capture markers. The transform between optical frame and the marker frame is computed on the 3D points extracted from the fiducial marker. The rest of the transformation chain is extracted via motion capture.

Fig. 3 depicts the calibration setup: a fiducial marker is placed on the floor, inside the Field of View (FoV) of the stereo camera Bumblebee2, which is mounted on the PTU. The fiducial marker is surrounded by MoCap markers, so that the mutual position between the frame origins of MoCap polygon and fiducial marker are known. The transform ${}_bT_c$ is then computed as:

$${}_bT_c = ({}_wT_b)^{-1} {}_wT_m ({}_cT_m)^{-1} \quad (1)$$

where ${}_wT_b$ and ${}_wT_m$ are the transforms — provided by the motion capture system — that project the base and the marker into the world frame, while ${}_cT_m$ is the transform between the marker and the camera frame of reference; its translational component \mathbf{t} is extracted directly as the coordinates of the marker center in the camera frame, while the rotational part is computed through Singular Value Decomposition as solution to the Wahba’s problem [15]:

$${}_mT_c = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} U M V^T & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad (2)$$

where M , U and V are extracted from:

$$B = \sum_{i=1}^n \mathbf{w}_i \mathbf{v}_i^T = U S V^T \quad (3)$$

$$M = \text{diag} [1 \quad 1 \quad \det(U) \det(V)] \quad (4)$$

and \mathbf{w}_i , \mathbf{v}_i are the sets of corners points on the marker coordinate frame and on the camera frame, respectively.

TABLE I
OUR SENSORS DETAILS.

	Output	Accuracy (m@m)
BB2	0.8 MP 20 fps	0.01@1.29, 0.5@9.6, 1@13.6
Xtion	0.3 MP 30 fps	0.001@0.8, 0.04@4

	Range (m)	Valid depth pixels	HFOV-VFOV (d)
BB2	0.6 - 20	65%	97 - 72
Xtion	0.8 - 4	84%	57 -40

B. Synchronization

The importance of synchronization between heterogeneous sensor samples is confirmed by the need of several groups to solve it via hardware, with custom FPGA boards [7], [16], [17]. Although this appears to be the most effective solution, it forces either the development of a custom hardware board or the purchase of new hardware. On the other hand, synchronization via software is less expensive, more flexible even though potentially less accurate.

To ensure that the processed IMU and depth signals are fused only when referred to the same time event, we used the commodity methods provided by the Robot Operating System (ROS).

C. Sensor noise and calibration

The noise of a point cloud highly depends on the sensor used. Tab. I shows an accuracy comparison between the two sensor we are equipped with. In this paper we process data from the Asus Xtion sensor as we considered an indoor environment. Even though it is more reliable than a stereo camera, when the distance approaches a few meters, the accuracy drop starts becoming non-neglectable. To cancel the noise few preprocessing can be used. For example, [18] recalibrated the structured light sensors to improve the depth image quality and [19] corrected depth distortion.

IV. DEPTH AND INERTIAL LOCALIZATION

In this section we describe how we improve the ICP registration and how we fuse it with inertial data.

A. Depth pre-processing and ICP registration

The Iterative Closest Point (ICP) is the most used method for point cloud registration [20], [21]. It has been demonstrated that its performance drops dramatically if the number of points that form 3D features is significantly smaller than sets of points that yield ambiguities (*e.g.*, planar surfaces) [22]. Furthermore, processing a full point cloud per frame is computationally expensive for a typical onboard computer.

Hence, for each couple of point clouds we aim at selecting the minimum subset of points that produce good geometrical features to estimate the robot’s motion, and discarding the rest. Good candidates for such selection are the points extracted from depth image pixels that change their intensity abruptly between two consecutive frames. If we exclude noise side effects, these are indeed the points that are likely to carry most of the information about both geometry and motion. The

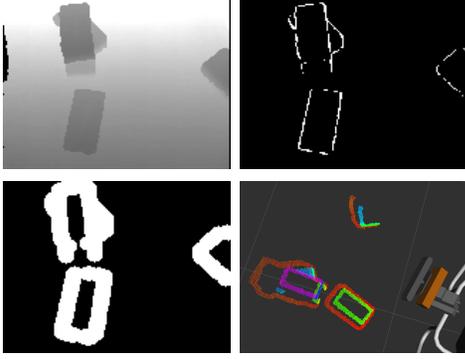


Fig. 4. Image preprocessing for point selection (from left to right, from top to bottom): current frame, image difference (thresholded), dilation, final result.

process of point selection we propose includes the following steps (see Fig. 4):

- **Background subtraction:** we compute the absolute difference of two consecutive frames. Brighter pixels are likely to correspond to moving 3D features (*e.g.*, the edge of a brick), while darker pixels to objects far from the optical center or to geometrically featureless areas (*e.g.*, the floor). The difference is then thresholded to have a black and white image (top right corner, Fig. 4).
- **Dilation:** we operate a morphological operation to expand the neighborhood of candidate points. The dilation size d is crucial for the performance: too large values are likely to produce big clouds with many featurless areas, whilst small values are likely to create variability between one cloud and the following.
- **Temporal fusion:** to avoid abrupt changes between clouds, we perform a bitwise OR of the current dilated frame with the previous k frames. Also k plays an important role for both speed and accuracy.

B. ICP registration and IMU fusion

To converge, ICP requires a good initial guess for the registration. Similarly to [6], we compute the guess from the IMU angular positions Q_i and linear accelerations a_i . In particular, the angular position differential is computed as the quaternion multiplication $Q_i \cdot Q_{i-1}^{-1}$, while the linear position differential is computed by double integration of the raw accelerations using the trapezoidal rule:

$$v_i = v_{i-1} + \frac{1}{2}(a_i + a_{i-1}) \Delta t \quad (5)$$

$$p_i = p_{i-1} + v_{i-1} \Delta t + \frac{1}{2}(\Delta t)^2 \left(\frac{2a_{i-1}}{3} + \frac{a_i}{3} \right) \quad (6)$$

where v_i and p_i are the current velocity and position, respectively.

The raw accelerations are first transformed into the same frame of reference and low passed filtered before computing the corresponding velocities and positions.

Given the IMU guess, the registration algorithm works as follows (see Alg. 1): first we compute the transform between

the two clouds C_i and C_{i-1} by means of ICP with guess T_i^{IMU} . Then, if the resulting transform T^{ICP} is valid, roll and pitch of T^{ICP} are replaced with the ones of T_i^{IMU} , which are more reliable because computed with the gravitational field. An ICP transform is valid if:

- 1) the ICP fitness score is below a threshold (1×10^{-4})
- 2) the roll and pitch estimated by the ICP are falling in an interval around the IMU roll and pitch.
- 3) the translation and rotation are smaller than the maximum threshold estimated empirically.

In case the transform is invalid, we proceed to dead reckoning with orientation taken from the IMU and position taken from the previous valid transform.

In case a global registration is needed, we update T_i^{global} , otherwise we transform the map with the current transform, as detailed in Section V.

Algorithm 1 Point Cloud Registration

```

1:  $i \leftarrow 1$ 
2:  $T_{\text{global}} \leftarrow I_4$ 
3:  $\text{getData}(C_0, Q_0, a_0)$ 
4:  $C_{\text{target}} \leftarrow C_0$ 
5: while  $\text{getData}(C_i, Q_i, a_i)$  do
6:    $T_i^{\text{IMU}} = \text{computeGuess}(Q_i, Q_{i-1}, a_i, a_{i-1})$ 
7:    $T_i^{\text{ICP}} \leftarrow \text{ICP}(C_{\text{source}}, C_{\text{target}}, T_i^{\text{IMU}})$ 
8:   if  $\text{isInvalid}(T_i^{\text{ICP}})$  then
9:      $T_i^{\text{ICP}} \leftarrow T_i^{\text{ICP}}$ 
10:  end if
11:   $\text{roll}(T_i^{\text{ICP}}) \leftarrow \text{roll}(T_i^{\text{IMU}})$ 
12:   $\text{pitch}(T_i^{\text{ICP}}) \leftarrow \text{pitch}(T_i^{\text{IMU}})$ 
13:   $T_i^{\text{global}} \leftarrow T_{i-1}^{\text{global}} T_i^{\text{ICP}}$ 
14:   $C_{\text{target}} \leftarrow C_{\text{source}}$ 
15:   $i \leftarrow i + 1$ 
16: end while

```

V. LOCAL MAPPING

Standard approaches to localization that are based on point cloud registration involve the collection of consecutive transforms between incoming clouds and their cumulative multiplication to get the transform between the robot base in the frame of reference of the first cloud, which is taken as origin of a fixed world frame.

Given a cloud C_n , where n is an index of both time and frame number, the recursive equation to get the map M_n is:

$$M_n = M_{n-1} + \prod_{i=0}^{n-1} T_{i+1} C_n \quad (7)$$

with the initial condition $M_0 = C_0$. The close form of Eq. 7 is:

$$M_n = C_0 + \sum_{i=0}^{n-1} \left(\prod_{j=0}^i T_{j+1} \right) C_{i+1} \quad (8)$$

Eq. 8 shows that as i increases (*i.e.*, we get closer to the most recent cloud) the expansion of the product between brackets includes more and more terms. Since each term

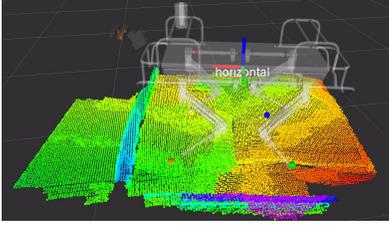


Fig. 5. Example of local mapping on a flat terrain with an horizontal obstacle. The rear part of the map is older and drifted on z axis

carries uncertainties, this ends up into an unavoidable drift because the localization is referred to a global and fixed world frame. Loop-closure detection algorithm are usually used to correct this drift.

However, a global localization is not needed for foothold planning, since events far away in space and time should not affect the decision for the steps to be taken in the near future. Hence, the localization accuracy should be proportional to the freshness of the data. Here we propose the following mapping equation:

$$M_n = C_n + {}_n T_{n-1} M_{n-1} \quad (9)$$

where the initial condition is again $M_0 = C_0$ and the close form is:

$$M_n = C_n + \sum_{i=0}^{n-1} \left(\prod_{j=n}^{n-i} T_{j-1} \right) C_{n-i-1} \quad (10)$$

Eq. 10 shows that the expansion of the product between brackets now has less and less terms when the index of the involved cloud approaches n .

Geometrically, the maps of Eq. 8 and 10 differ by a rigid transform, because in the first case the map is composed by clouds aligned to the first sample, while in the second all the clouds are expressed in the frame of the last one.

Hence, the drift is the same for both maps, but the local map has the advantage of pushing the drift error away in time, which is more convenient when performing navigation.

An example of local mapping is depicted in Fig. 5.

VI. EXPERIMENTAL RESULTS

To show the effectiveness of our SLAM system, we performed an indoor trotting sequence with a motion capture system as ground truth. Even though we use local frame for mapping, we performed global registration to compare the localization performance against the motion capture system.

The robot was tele-operated in order to cross different obstacles arranged on a flat surface. Depth images and point clouds were recorded at 15 Hz and IMU data at 250 Hz. Then, we replayed the sequence on the same machine (Intense PC2) at the same rate and computed the global localization with the ICP alone and with our proposed algorithm (Alg. 1).

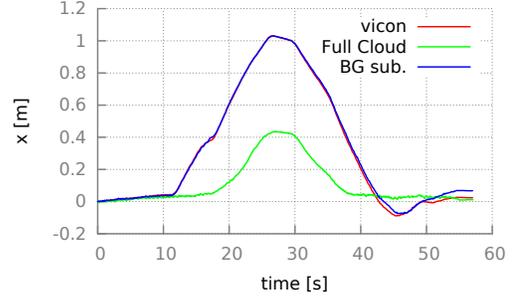


Fig. 7. Improvement of the ICP using the cloud segmentation based on depth background subtraction.

A. Background subtraction and Full cloud

An example for the x -axis of the improvement carried out by the background subtraction against full cloud is depicted on Fig. 7. The segmentation improves the tracking by more than 100% for the central part of the run, where the full cloud, having more static points than good features makes the ICP underestimate the motion. The background subtraction reduces the number of points by a factor of 10, allowing to operate within the real-time constraint. In contrast, for the full cloud we had to reproduce the data log two times slower.

B. Inertial-ICP and ICP

Fig. 6 shows the estimation of the robot global position and orientation while the robot was trotting forward and backward in the area covered by the motion capture system ($\sim 3 \text{ m}^2$).

Both orientation and rotation are closer to the ground truth if compared with the ICP only version. As expected, the major improvement is given by the pitch and roll, which are directly taken from the IMU. In both methods a non-neglectable drift in the z -axis is visible. This is due to the lack of features in the zy -plane of the robot, since the camera was facing the ground and therefore the estimated motion was more accurate on the xy -plane.

With our method the maximum drift after 60 seconds of trotting is about 10 cm on the y -axis, corresponding to a 5% of the total path. It has to be noted that the robot was often changing direction on the yaw axis, making the estimate of y more difficult than other axes.

VII. CONCLUSIONS AND FUTURE WORK

Robot locomotion highly depends on the ability to perceive and map the environment. In this paper, we presented a localization method that uses point cloud registration with inertial measurements for real-time local mapping. Our local SLAM method is robust and fast enough to produce consistent maps of the The SLAM system has been implemented onboard our robot HyQ and assessed by using a motion capture system as a ground truth. In our future work, we plan: a) to assess the framework in an outdoor environment using the point cloud provided by our stereo camera (already mounted on the robot); b) to integrate the real-time localization and mapping to our reactive trotting or planned locomotion framework, in

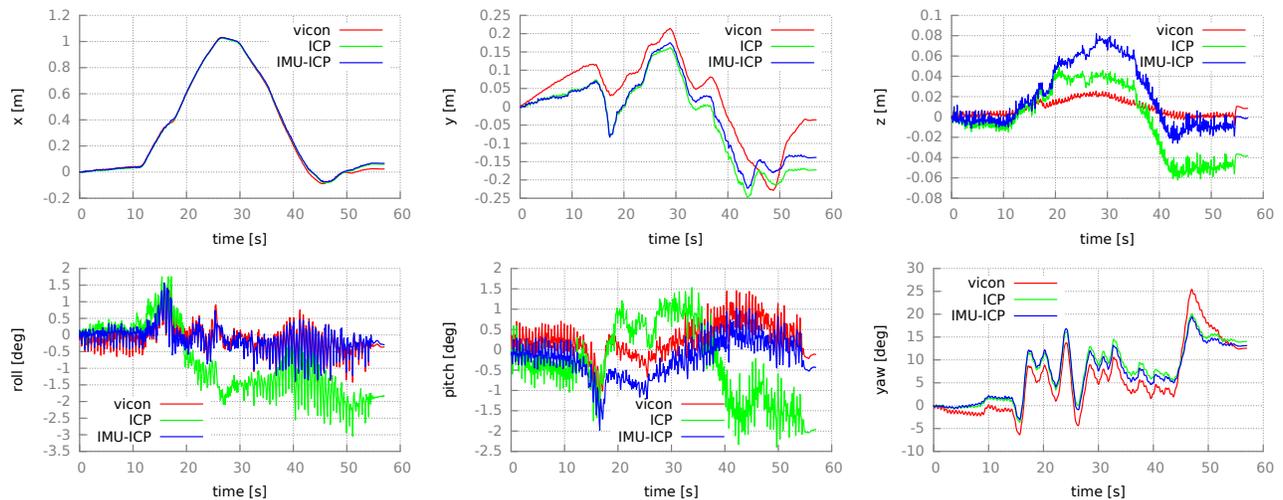


Fig. 6. Experiment with the real robot trotting indoor. The six graphs represent the 6 DOF pose of the robot in the global frame of reference estimated by the motion capture system (red), the ICP (green) and our framework (blue).

order to perform reactive behaviours in a fully autonomous manner.

ACKNOWLEDGMENT

This work was supported by Istituto Italiano di Tecnologia.

REFERENCES

- [1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *Robotics Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99–110, 6 2006.
- [2] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (slam): part ii," *Robotics Automation Magazine, IEEE*, vol. 13, no. 3, pp. 108–117, 9 2006.
- [3] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of HyQ - a hydraulically and electrically actuated quadruped robot," *J. of Systems and Control Engineering*, 2011.
- [4] A. Winkler, I. Havoutis, S. Bazeille, J. Ortiz, M. Focchi, R. Dillmann, D. G. Caldwell, and C. Semini, "Path planning with force-based foothold adaptation and virtual model control for torque controlled quadruped robots," in *IEEE ICRA*, 2014.
- [5] S. Bazeille, M. Camurri, J. Ortiz, I. Havoutis, D. G. Caldwell, and C. Semini, "Terrain mapping with a pan and tilt stereo camera for locomotion on a quadruped robot," in *ICRA14 Workshop on Modelling, Estimation, Perception and Control of All Terrain Mobile Robots (WMEPC14)*, 2014.
- [6] M. Nießner, A. Dai, and M. Fisher, "Combining inertial navigation and icp for real-time 3d surface reconstruction," *EG*, 2014.
- [7] K. Schmid and H. Hirschmuller, "Stereo vision and imu based real-time ego-motion and depth image computation on a handheld device," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 4671–4678.
- [8] U. Qayyum and J. Kim, "Inertial-kinect fusion for outdoor 3d navigation," in *Australasian Conference on Robotics and Automation 2013 (ACRA 2014)*, 12 2013.
- [9] A. Bethencourt and L. Jaulin, "3d reconstruction using interval methods on the kinect device coupled with an imu," *International Journal of Advanced Robotic Systems*, vol. 10, no. 93, pp. 1–10, 2013.
- [10] J. Z. Kolter, K. Youngjun, and A. Y. Ng, "Stereo vision and terrain modeling for quadruped robots," in *IEEE ICRA*, 2009.
- [11] a. Stelzer, H. Hirschmuller, and M. Gerner, "Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain," *The International Journal of Robotics Research*, vol. 31, no. 4, pp. 381–402, 2 2012.
- [12] J. Ma, S. Susca, M. Bajracharya, L. Matthies, M. Malchano, and D. Wooden, "Robust multi-sensor, day/night 6-dof pose estimation for a dynamic legged vehicle in gps-denied environments," in *IEEE ICRA*, May 2012, pp. 619–626.
- [13] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Y. Siegwart, "Robot-centric elevation mapping with uncertainty estimates," *Proceedings of CLAWAR*, 2014.
- [14] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.
- [15] F. L. Markley, "Attitude determination using vector observations and the singular value decomposition," *The Journal of the Astronautical Science*, vol. 36, no. 3, pp. 245–258, 1988.
- [16] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. Furgale, and R. Siegwart, "A synchronized visual-inertial sensor system with fpga pre-processing for accurate real-time slam," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 5 2014, pp. 431–437.
- [17] G. Camellini, M. Felisa, P. Medici, P. Zani, F. Gregoretti, C. Passerone, and R. Passerone, "3dv — an embedded, dense stereovision-based depth mapping system," in *Proc. IEEE Intelligent Vehicles Symposium 2014*, Dearborn, MI, USA, 6 2014, pp. 1435–1440.
- [18] C. Nguyen, S. Izadi, and D. Lovell, "Modeling kinect sensor noise for improved 3d reconstruction and tracking," in *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, Oct 2012, pp. 524–530.
- [19] A. Teichman, S. Miller, and S. Thrun, "Unsupervised intrinsic calibration of depth sensors via slam," in *Robotics: Science and Systems*, 2013.
- [20] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *Proc. of the 2011 10th IEEE Int. Symp on Mixed and Augmented Reality*, ser. ISMAR '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 127–136.
- [21] M. Bosse, R. Zlot, and P. Flick, "Zebedee: Design of a spring-mounted 3-d range sensor with application to mobile mapping," *Robotics, IEEE Transactions on*, vol. 28, no. 5, pp. 1104–1119, Oct 2012.
- [22] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *Third International Conference on 3-D Digital Imaging and Modeling (3DIM 2001)*, 2001, pp. 145–152.