Distributed Multi Sensor Data Fusion for Autonomous 3D Mapping

Jose E. Guivant, Samuel Marden and Karime Pereida
School of Mechanical Engineering
UNSW
Sydney, Australia
j.guivant@unsw.edu.au ; s.marden@unsw.edu.au ; k.pereidaperez@student.unsw.edu.au

***Abstract*— Indoor and outdoor accurate 3D mapping is a relevant resource for a diversity of applications. This paper describes an autonomous platform capable of generating 3D imagery of the environment in unknown indoor and outdoor contexts. The system is composed by a number of Data Fusion processes that are performed in real-time by on-board and/or off-board processing nodes. The platform's sensing capabilities are composed of multiple laser scanners for 2D and 3D perception, IMU units, 3D cameras (indoor Kinect), standard cameras, GPS (for outdoor operation) and dead reckoning sensors. The acquired data is shared with multiple client processes that are in charge of different levels of perception and control. The resulting data, produced by the perception processes is also shared for being used by higher-level processes such as the 3D mapping, generation of maps of diverse dense properties, detection and classifications of obstacles and other context features that are application specific.**

Real-time Distributed Sensor Data Fusion; Autonomous mapping. 3D mapping. Scan Matching.

I. INTRODUCTION

The modeling of dense representations of the environment requires a set of resources such as diverse sensing capabilities, sensor data fusion processes, localization and processes for sharing the information with the consumers of the generated information. This paper describes a platform (and its associated system) that is able to autonomously generate 3D maps of indoor and outdoor contexts. A human operator or even a high level process can specify destination points that the platform must reach. In order to reach the specified destinations, the platform must understand the context of operation by generating a description of it, e.g. through a 3D map (and other dense representations if proper sensing capabilities are available). Based on the last updated belief about the environment the platform is able to maintain an optimal plan and to follow it. As the generated belief is periodically updated the planning is also performed periodically in order to maximize the probability of achieving the optimal plan by considering the last available belief provided by the perception processes. As the platform learns the environment through its trip to the destination a detailed 3D map of the environment is generated.

In order to describe this system (or "system of systems") it needs to be explained by describing its more relevant components. Those include the following subsystems:

1) Low Level Perception: Sensing.
2) Medium Level Perception:
   - 2.1) Short Term 3D localization
   - 2.2) Data Fusion for 3D imagery
   - 2.3) 3D Long Term Localization based on Scan Matching
   - 2.4) 2D Long Term Localization based on SLAM
3) High Level Perception:
   - 3.1) Traversability Maps,
   - 3.2) Terrain Classification
4) Planning Process
5) Distributed Processing and Data Base Service.

Item 1 corresponds to the sensing capabilities of the platform. Although the set of sensors is scalable, a typical set of sensors are described in section II. It is also important to emphasize the quality of the measurements, in particular the accuracy of the timestamps of the measurements, as those are relevant for certain data fusion processes. A description of the high accuracy estimation of timestamps is presented in section V.

Item 2 focuses on the synthesis of estimates that are needed by higher-level processes. Pose (position and attitude, in 6DoF) estimates that are called "short term" correspond to estimates that must be accurate in relative terms, i.e. for describing relative variations of the platform pose during short intervals of time (e.g. a few seconds). Those estimates are not intended for use in global localization, but rather for accurately knowing pose variations necessary for the synthesis of 3D imagery and by other client processes.

As the platform is not retrofitted with real 3D sensing capabilities, the 3D images are generated with data provided by a standard 2D laser scanner. For that purpose one of the

978-1-4673-1954-6/12/$31.00 ©2012 IEEE

Authorized licensed use limited to: Xi'an Jiaotong-Liverpool University. Downloaded on April 02,2022 at 06:43:42 UTC from IEEE Xplore. Restrictions apply.

platform's 2D laser scanners is used in a "3D" fashion, i.e. the sensor is mechanically rotated to cover solid angular regions allowing the synthesis of 3D images. The whole fusion process is performed in real time when the platform is travelling (i.e. it is usually not stationary). This process combines information provided by three data sources: the short term estimates of the platform's pose, the relative rotations of the laser scanner and the laser scanner measurements. This process is described in section III.

The estimates provided by (2.1) and (2.2) are used by a scan matching process that performs the platform's global localization. The scan matching process can be seen as an accurate 6DoF dead-reckoning process, for global localization. It is highly accurate but it is an incremental process that can accumulate error in long-term distances, however it is definitely more accurate than using the short term estimates for global localization. It is used in place of 2D SLAM in contexts such outdoor, where the terrain is not flat (what makes the 2D SLAM inadequate). The scan matching approach is described in section IX.

Finally, for indoor cases, where the terrain is flat (and the platform kinematics is almost constrained to 3DoF), a 2D SLAM process can be used. This component is not discussed in this paper.

The results generated by components in (2) are usually used as input data for higher-level processes. Although those consumer processes are usually not a part of the core system, some core subsystems do need the 3D imagery as well. The platform needs to plan and the plans need to be synthesized based of the description of the context of operation. As the system itself is responsible for generating and maintaining a description of the environment, the 3D imagery is exploited for that purpose. For this reason item (3) is included in this paper and discussed in section VIII. That section explains how the 3D imagery is used for generating a "Traversability Map". The 3D information is used to generate features of interest for modeling the terrain, particularly focused on describing and classifying the terrain according to the degree of difficulty or risk for the platform. This class of information can be compressed to lower dimensionality, i.e. through 2D discrete maps, with Occupancy Grids ([11],[2]) being a particular case. This level of perception may be considered to be between medium and high-level perception.

Based on the availability of a fresh (updated) representation of the environment, in particular regarding the characteristics of the terrain relevant for the transit of the platform, it is feasible to perform a planning process that is based on the description of the environment. Its goal is the generation of a path that satisfies the main objective of reaching a destination (or a set of destinations). Naturally, we expect the platform to satisfy that requirement by investing the lowest cost (distance, time, energy) and minimizing risk, e.g. collisions with obstacles, platform stress, pot-holes, etc). In order to solve this problem an optimal planner of the Dynamic Programming family is applied in real time. Section X briefly describes the planning process.

This entire set of processes is necessary for achieving the high level goal of generating a belief of the area of operation that can be exploited by consumer processes. Those client processes do not usually run "on-board", i.e. on the robot's computer. Instead they usually operate on remote processing nodes, distributed through a local network or even remote nodes connected through the internet. The process for sharing the data resources is presented in the section VI. The Data Replication allows remote processes to read and publish data in the distributed Data Base. The way in which the information generated by the platform is shared through the communication resources is transparent for the client processes.

II. SENSORS

This section lists the typical set of sensors used in the platform operation, which includes 2D laser scanners, inertial units (IMU), indoor 3D cameras (Kinect) and other sensors.

A. Laser Scanners

Three laser scanners (two of the family LMS151 and one LMS200) are installed on the current configuration of the platform. The LSM151 units are operated at 50Hz and the LMS200 at 38Hz. Two of the units are used in a usual 2D mode, i.e. they were installed scanning in a horizontal plane, one facing ahead the platform and the other facing behind it, with the goal of being used for 2D localization and other "2D" purposes. A third laser scanner is used in "3D mode", i.e. for producing 3D images of the platform's surroundings. The description of the 3D imagery is presented in section III.

B. Inertial Measurement Units (IMU)

Two 3D IMU units provide 3D accelerometers, 3D gyros and magnetometers. Both units are usually operated at a frequency of 200Hz. The units are of different quality (low and high cost).

C. Dead Reckoning

Dead reckoning measurements involve wheels and steering encoders. An additional encoder is used for measuring the position of the electric motor that operates rotating laser (3D scanner).

D. Vision

A Kinect camera provides images which are acquired at a frequency of approximately 5 Hz with a resolution of 640 x 480 pixels for both 24 bits RGB and 11 bits depth.

Virtual Sensors: In addition to the real sensors, a number of "virtual sensors" are generated by the on-board cpu. Those measurements are:

1) Vehicle's short-term 6DoF pose estimates (by fusion of IMU measurements and wheel encoders)
2) 3D laser images (by fusion of laser scanner, motor encoder and the vehicle's 3D pose estimates)
3) 2D global pose estimates provided by a 2D SLAM (Simultaneous Localization and Mapping) process.
4) Long term 6DoF pose estimates (by scan matching of 3D images)

Authorized licensed use limited to: Xi'an Jiaotong-Liverpool University. Downloaded on April 02,2022 at 06:43:42 UTC from IEEE Xplore. Restrictions apply.

Figure 1. A picture of The platform. A LMS2151 laser scanner can be seen at the top of the platform.

III. 3D SYSNTHESIS

In this work the 3D laser scanning capability is achieved through fusion of measurements provided by a rotating laser scanner and 3D pose estimates of the moving platform. As the fusion process is performed in real time by the on-board computer, the generated 3D frames are offered as a "virtual sensor" to the client applications.

A laser scanner (model LMS151), configured to scan 270 degrees at a resolution of ½ degree, is rotated in order to scan a solid angle of 270 by 180 degrees. The rotating laser scanner, installed on the top of the platform, can be seen in Fig. 1 and a sketch of its operation in Fig. 2.

A detailed description of the fusion process that generates the 3D images by processing the laser scanner measurements, the encoder of the rotating motor and the 6DoF short term estimates, can be found in [7], [8] and [10].

A. 3D Laser Scan Lines and 3D Frames

The 3D data is composed of a cloud of 3D points. There is certain order in the sequence of points due the nature of the acquisition and fusion processes. Each laser scan provided by the laser scanner is projected to 3D through fusion of the scan's ranges (and their implicitly associated angles in the scanning plane), the platform's 3D pose and the azimuth position of the rotating motor. The motor follows a triangular movement, i.e. a ramp from the minimum angle to the maximum one followed by another ramp in opposite direction; these ramps are repeated periodically. This periodic angular scanning generates 3D images having a field of view of approximately 270 degrees in elevation and a range $[\alpha_{\min}, \alpha_{\max}]$ in azimuth, respect to the platform coordinate frame.

The angular speed and amplitude of the ramps are defined by higher-level applications, depending on the purpose of the 3D imagery. Increasing the frequency of the ramp implies an increase in frame rate but a decrease in image resolution (density of points). The angular interval $[\alpha_{\min}, \alpha_{\max}]$ can also be set, thereby altering the resulting field of view.

For mapping and control purposes a compromise was taken, resulting in an adequate resolution, field of view and frame rate of the generated 3D images.

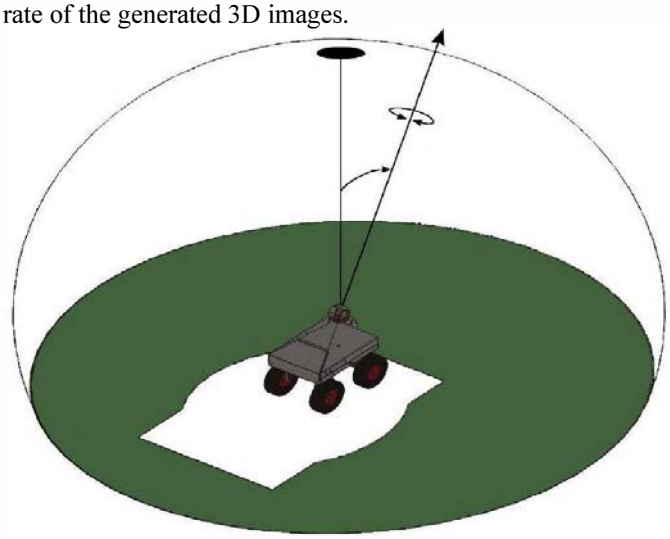

Figure 2. Field of view of the 3D system, observed from the top.(Figure extracted from [8]).

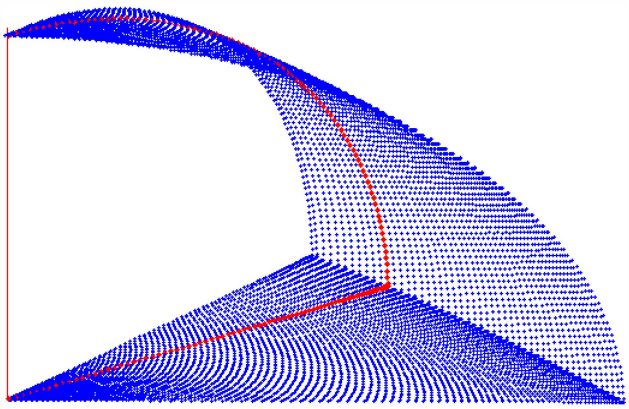

Figure 3. Approximated field of view of the 3D frame for a rotating ramp covering 25° in azimuth. The set of red points corresponds to an individual 3D scan.

The density of points is approximately 0.5 degrees in elevation and $\delta\alpha \cong \omega_{\beta} \cdot 1/50$ in azimuth, where ω_{β} is the angular velocity of the rotating motor (during the linear sections of the ramps). The factor 1/50 is due to the sensor scan rate (50 frames/second). The 0.5 degrees of elevation angle resolution is due to the resolution of the laser scanner (0.5 degrees). The resulting nominal field of view can be seen in figures 2 and 3.

Although the set of points is just a sequence of points, it is convenient to organize them as 3D scans ("scan lines") or as 3D images called 3D frames. A 3D laser scan is just a 2D laser scan projected to 3D. A 3D frame is composed by a sequence of 3D laser scans that correspond to the interval of time when

Authorized licensed use limited to: Xi'an Jiaotong-Liverpool University. Downloaded on April 02,2022 at 06:43:42 UTC from IEEE Xplore. Restrictions apply.

the rotating motor is moving from one extreme to the opposite (one ramp).

A 3D frame is an image that can even be parameterized as a 2D surface that is function of two parameters: azimuth and elevation angles. In Fig. 4 a typical 3D frame is shown. The image was one of the many that were taken during an experiment when the platform was moving.

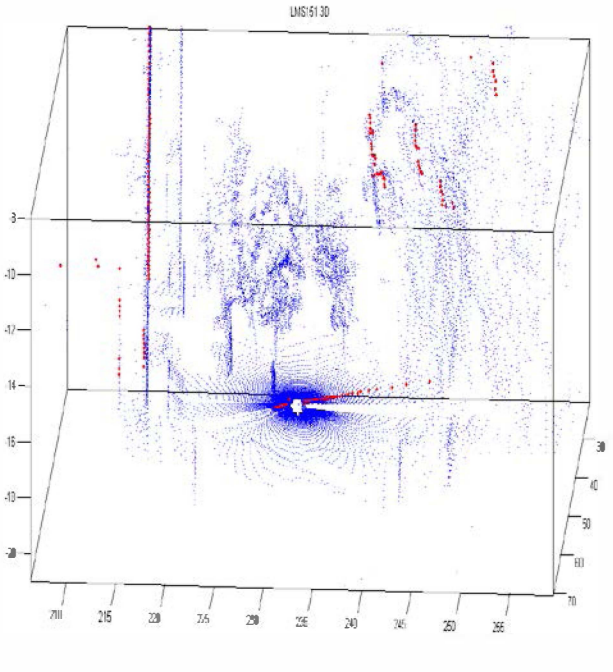

Figure 4. A 3D frame (blue points) and an individual 3D scan line (red points). The trunk and canopies of trees can be inferred from the image. The image corresponds to a 3D frame covering 180 degrees in azimuth.

IV. Dead Reckoning Estimates

In order to generate high quality 3D imagery the fusion process needs to use high quality estimates of the platform 3D pose. As each 3D frame is composed of data collected during a short period of time (e.g. 3 seconds), the fusion process only requires 3D pose estimates of high accuracy in relative terms, i.e. it does not require global localization for that purpose. Pose estimates are 6DoF, i.e. 3D position and 3D attitude that are provided by an on-board dead reckoning fusion process. Angular velocities (from the IMU's gyros) and speed (wheel encoder) are used in the dead reckoning process that also considers kinematic constraints, as presented in [1]. This constraint assumes that at a defined point on the platform the nominal velocity and attitude vectors are parallel.

The attitude and the offsets of the gyros are estimated by an EKF estimator that updates the estimates through the observation of attitude increments, e.g. the ones provided by the scan matching process. The process model for the 3 attitude angles considers that the measured angular rates are polluted by the unknown offsets and white noise. The process model is expressed as follows,

$$\begin{aligned}
\frac{d\varphi_x}{dt} &= \omega_x(t) + \left(\omega_y(t)\cdot\sin(\varphi_x(t)) + \omega_z(t)\cdot\cos(\varphi_x(t))\right)\cdot\tan(\varphi_y(t)) \\
\frac{d\varphi_y}{dt} &= \left(\omega_y(t)\cdot\cos(\varphi_x) - \omega_z(t)\cdot\sin(\varphi_x)\right) \\
\frac{d\varphi_z}{dt} &= \left(\omega_y(t)\cdot\sin(\varphi_x) + \omega_z(t)\cdot\cos(\varphi_x)\right)\Big/\cos(\varphi_y)
\end{aligned} \quad (1)$$

Where the perfect (but unknown) local angular rates, $\omega_x, \omega_y, \omega_z$ can be expressed as a linear combination of the measured angular rates ($\tilde{\omega}_x, \tilde{\omega}_y, \tilde{\omega}_z$), the offsets ($b_x, b_y, b_z$) and white random noise ($\zeta_x, \zeta_y, \zeta_z$), as follows

$$\begin{aligned}
\omega_x(t) &= \tilde{\omega}_x(t) - b_x + \zeta_x(t) \\
\omega_y(t) &= \tilde{\omega}_y(t) - b_y + \zeta_y(t) \\
\omega_z(t) &= \tilde{\omega}_z(t) - b_z + \zeta_z(t)
\end{aligned} \quad (2)$$

The offsets are assumed to be slowly time varying, modeled by the following process model,

$$\frac{db_x}{dt} = 0 + \mu_x(t), \quad \frac{db_y}{dt} = 0 + \mu_y(t), \quad \frac{db_z}{dt} = 0 + \mu_z(t) \quad (3)$$

Where the uncertainty components in the process model (μ_x, μ_y, μ_z), are assumed to be white noise.

It must be noted that the updates only affect the estimates after each 3D frame is synthesized, i.e. during the interval when a 3D frame is being acquired and generated the estimation process only performs prediction steps, which are estimated using the last estimated offsets, and updated based on the previous observations.

It can be seen that there is a critical mutual dependency between two fusion processes. The accuracy of the attitude estimates is highly dependent on the estimates of the gyros' offsets that are corrected thanks to the observations provided by the scan matching process. Concurrently, the accuracy of the scan matching process is also highly dependent on the quality of the 3D frames, which can be distorted if the short-term estimates of the attitude are inaccurate. This "chicken-and-egg" dependency implies that certain proper initialization step is recommended for a fast convergence. For this reason, the full estimation process is initialized by obtaining the initial values of the offsets based on measurements of the angular velocities having the platform stationary for an interval of time (e.g. 5 seconds). In this way, the 3D images are never distorted due to inaccurate short-term estimates of the attitude.

Authorized licensed use limited to: Xi'an Jiaotong-Liverpool University. Downloaded on April 02,2022 at 06:43:42 UTC from IEEE Xplore. Restrictions apply.

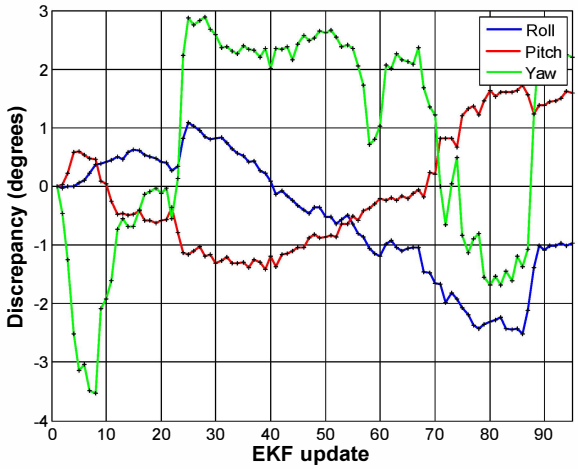

Figure 5. Absolute discrepancy between pure dead-reckoning attitude estimates and estimated attitude based on updates provided by the scan matching process.

Fig. 5 shows the difference between the dead-reckoning attitude and the attitude estimated by the scan matching process. Fig. 7 shows the convergence of the estimates of the gyros' offsets. The top subfigure show the estimated offsets using an off-line approach, which generates a PWL approximation based on the offsets evaluated when the platform was stationary. The lower subfigure shows the offsets estimated in real time, provided by the observer. The two fusion processes, the pose and offsets estimation and the scan matching, were found to be able to be performed concurrently, whose good combined performance can be corroborated by the accurate final 3D maps, in contrast with the 3D maps generated based on pure dead-reckoning, irrespective or whether or nor this process uses proper offsets.

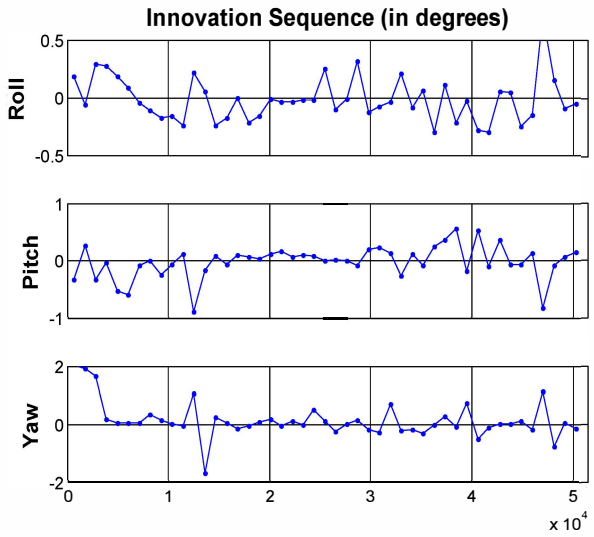

Figure 6. Innovation sequence, i.e. discrepacy between estimated relative attitude and measured .relative attitude provided by the scan matching process. The scale in the time axis is expressed in samples of the IMU sensor (5ms sample rate). Estimator updates are performed at low frequency, every two 3D frames (approximately 5.4 seconds).

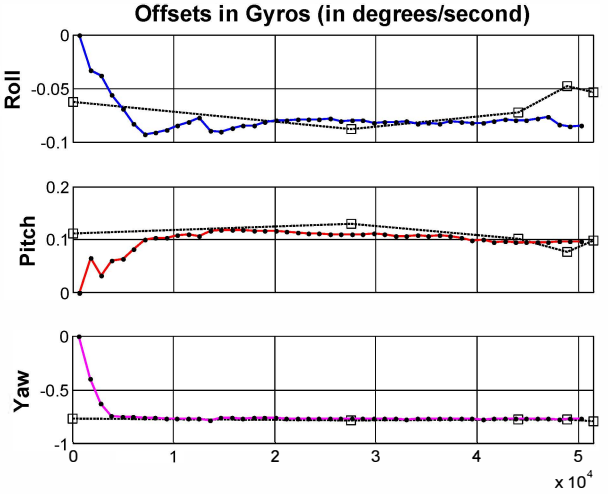

Figure 7. Estimates of the gyros's offsets for the on-line EKF estimator (continuous line and dots) and for the PWL off line estimates (dashed line), for the roll, pitch and yaw. All the offsets are expressed in degrees/second. In order to show the worst case scenario, the offsets were initialized to be zero.

V. ESTIMATION OF TIMESTAMPS

The performance of data fusion processes is affected by the accuracy of the measurements. This involves not just the accuracy of the measurements themselves but their associated timestamps as well.

For an individual sensor whose measurements are to be used in a data fusion process, each measurement must have an associated timestamp, expressed in the system's clock as $t^{(A)}$. The system's clock is the common clock; this common time reference allows proper data fusion of multiple sensors. There is a relation between the time expressed according to the system's clock and the individual sensor's clock

$$t^{(A)} - t_0^{(A)} = c \cdot \left(t^{(B)} - t_0^{(B)} \right) \Rightarrow t^{(A)} = c \cdot t^{(B)} + \beta \qquad (4)$$

Where the measurement's timestamp $t^{(A)}$ is referred to the clock A (system) and $t^{(B)}$ is the timestamp according to the sensor's internal clock.

If both clocks' frequencies are stable the parameters $(C\ ,\ \beta)$ are then constant. If some drift (in at least one of the frequencies) does happens then the parameters $(C\ ,\ \beta)$ should be assumed to be slowly time varying.

Depending on the type of sensor the time $t^{(B)}$ can be well known, partially known or completely unknown, as described in the following classification:

Case 1) $t^{(B)}$ is known (provided by the sensor as part of the measurement data). This case is the simplest one. The estimator just estimates the constants (or slowly time varying variables) C, β.

Case 2) $t^{(B)}$ is unknown but some characteristics are known.

2.1) $t_{k+1}^{(B)} - t_k^{(B)} = \tau$ with τ constant (but unknown or approximately known).

Authorized licensed use limited to: Xi'an Jiaotong-Liverpool University. Downloaded on April 02,2022 at 06:43:42 UTC from IEEE Xplore. Restrictions apply.

2.2) $t_{k+1}^{(B)} - t_k^{(B)}$ unknown but of the form $t_{k+1}^{(B)} - t_k^{(B)} = N_k \cdot \tau$ where N_k is an unknown integer and τ is an unknown real constant. The integer N_k usually belongs to a small set of hypotheses.

Case 3) $t^{(B)}$ and $t_{k+1}^{(B)} - t_k^{(B)}$ are unknown and are not assumed to be constrained by any rule.

We consider an estimator for solving the cases (1), (2.1) and (2.2). The constraint expressed in cases (1) and cases (2.1) and (2.2) can be interpreted the process models in an estimation process and the measured timestamps as the observations.

Case (3) cannot be solved in such a way due to the lack of a "process model" (consequently the estimated timestamp is the just the measured timestamp).

For the feasible cases we assume that when we measure the acquisition time, in the acquisition node, there exists uncertainty mainly due to the latency of the system. The error is modeled through the random variable η_k.

$$\tilde{t}_k^{(A)} = t_k^{(A)} + \eta_k = c \cdot t_k^{(B)} + \beta + \eta_k \qquad (5)$$

The times $t_k^{(A)}$ and $\tilde{t}_k^{(A)}$ are the real and the measured timestamp respectively, according to the system's clock. The probability density function (PDF) of this source of uncertainty has the following characteristic.

$$p(\eta) = 0 \quad \forall \eta < 0 \qquad (6)$$

We also assume the random variables $\{\eta_k\}$ are independent and identically distributed.

Some extra information about the uncertainty's PDF can be considered, e.g. $p(\eta) = 0 \quad \forall \eta > \eta_{\max}$.

or a more relaxed constraint such as an exponential shape.

By applying Bayesian estimation the "enhanced" timestamps are estimated. The differences between the estimated and measured timestamps are shown in the examples presented.

For a case of the type (1) the estimator just estimates the parameters C and β. The timestamp $t_k^{(A)}$ is then estimated by applying the linear expression $\hat{t}_k^{(A)} = \hat{c} \cdot t_k^{(B)} + \hat{\beta}$, where $\hat{c}, \hat{\beta}$ are the estimated parameters.

For allowing frequency drifts the process model for the parameters is assumed to be polluted with uncertainty

$$\frac{dc}{dt} = 0 + \eta_c, \quad \frac{d\beta}{dt} = 0 + \eta_\beta \qquad (7)$$

Where the uncertainty components η_c, η_β are assumed Gaussian, zero mean and uncorrelated and with very low standard deviation. This process model allows the estimator to estimate very slow variations of the relative frequency and phase between clocks.

For cases of the type (2.1) and (2.2) the estimator also estimates $t_k^{(B)}$ based on its assumed periodicity.

Case (2.1) takes advantage of the assumption $N_k = 1$. This case is similar to case 1, because a counter can be implemented and used as a clock. This assumption is broken if some measurement message is lost and the acquisition node is not aware of the fact. If such situation can arise, then case (2.2) must be considered.

Case (2.2) is less observable due to the multi-hypothesis condition given by the introduction of an extra component in the vector of estimates, i.e. the integer random variable (r.v.) N_k. In this case the estimator must estimate three random variables, C, β, N_k. The variable N_k is constrained by the assumption that it belongs to a small set of hypotheses. A better estimation can be done by applying a smoothing approach (e.g. by increasing the dimensionality of the vector estimates). This means that the estimation of the current timestamp can be improved by future timestamps measurements. This approach does have sense if the estimated variables can be used with certain delay or in an OFF-LINE fashion. In such a case the estimated state would become $\left(c, \beta, \{N_i\}_{i=k-H}^{k}\right)$ where the horizon H is short enough to avoid the need of assuming time varying parameters c, β during that horizon of time. The estimation process generates a belief $p\left(c, \beta, \{N_i\}_{i=k-H}^{k} \mid K\right)$ that represents the random variable $\left(c, \beta, \{N_i\}_{i=k-H}^{k}\right)$ based on observations up to time k. In practical terms multiple versions of the estimates can be offered, in particular for the cases $H=0$ (no smoothing) and for certain $H>0$ (smoothing).

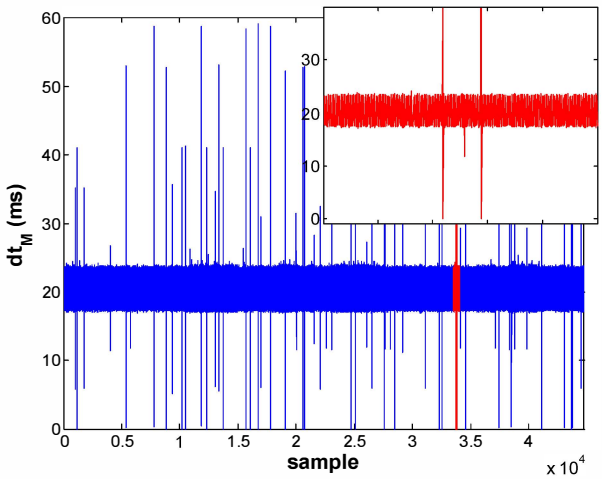

Figure 8. Interval between measured frames from a LMS151 laser scanner connected through an Ethernet link, according to the computer clock, raw timestamps. ($\Delta t_k = \tilde{t}_{k+1}^{(A)} - \tilde{t}_k^{(A)}$) The inset shows a zoomed interval.

A. *Experimental Results in Timestamp Estimation*

In this section the validation and correction of the timestamps associated to measurements provided by a LMS151 laser scanner are shown.

The sensor operates at approximately 50Hz. The measurements are acquired through an Ethernet link. As the network controller is heavily loaded attending other

Authorized licensed use limited to: Xi'an Jiaotong-Liverpool University. Downloaded on April 02,2022 at 06:43:42 UTC from IEEE Xplore. Restrictions apply.

communication purposes and the node's CPU is also loaded with a number of processing tasks, the measured timestamps are affected by latencies. Fig. 8 shows that the measured sampling interval is far from constant. The real sample interval must be periodic, i.e. 20 ms, with a deviation in the order of a fraction of a millisecond. In this case the estimator is able to achieve that performance by estimating the related parameters $\hat{c}, \hat{\beta}$. The discrepancy between measured and estimated timestamps, $\tilde{t}_k^{(A)} - \hat{t}_k^{(A)}$, is shown in Fig. 9. A relevant number of latencies were higher than 5 ms, even in some isolated cases where the latencies reached more than 25 ms. These latencies, if not estimated, would have relevant effect in the synthesis of the 3D images.

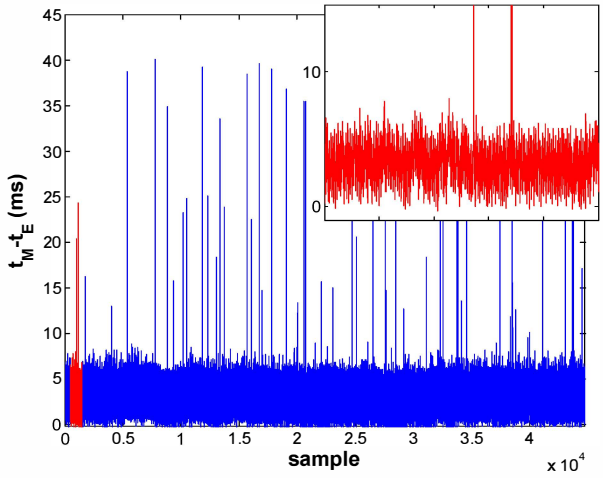

Figure 9. Residual between estimated timestamp and measured timestamps ($\tilde{t}_k^{(A)} - \hat{t}_k^{(A)}$), in milliseconds, for the LMS151 measurements. Note that the majority of the latencies are in the range between 0 to 7 ms. A number of the measurements show an excessive latency up to 40 ms.

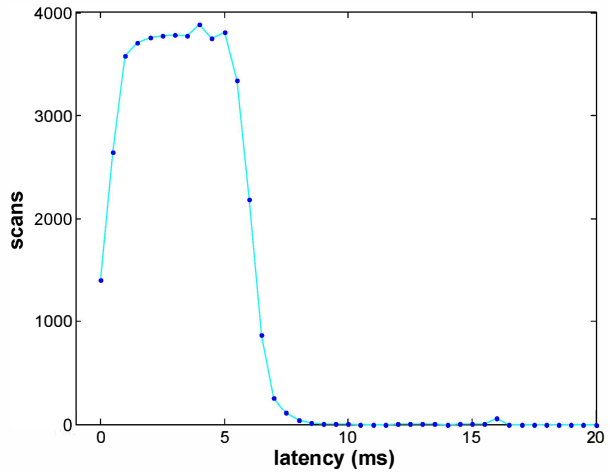

Figure 10. Histograms of the residual between estimated timestamp and measured timestamps ($\tilde{t}_k^{(A)} - \hat{t}_k^{(A)}$), in milliseconds, for LMS151 measurements.

VI. Data Replication

In this system the data fusion processes are capable of being performed in a distributed fashion, on nodes geographically distant. Although all the processes can be run on-board the platform there are cases where, in order to provide redundancy or to test more sophisticated algorithms, those can be performed in remote nodes. In order to allow such capability certain components of the data need to be shared through the network.

One of the characteristics of this software system is that the data resources are implemented as a database. The database allows client processes to publish data and to read published data in a real-time fashion. Processes can be located in different nodes.

The database is configured by clients to "replicate" certain items at certain nodes.

For example: A process that runs in node A reads a laser scanner connected to that node. It acquires 50 scans per second (~53kBytes/second). The process also publishes each new read scan in a database item. The database is configured to replicate that item, to be available at nodes B and C. At node B a process periodically reads that database item for new scans. It uses the data to infer the proximity of objects to the sensor.

In this scheme none of the processes care about the delivery of the data – all of the communication and message passing matters are transparent for the client processes. It is only the Data Base service that cares about those matters.

This scheme can be scaled to a high number of data resources and processes running in a set of nodes connected by multiple networks.

In this particular application there were usually 2 or 3 nodes. One node was the UGV itself. This node runs the low level acquisition and control processes.

A remote node, implemented through a desktop computer is able to receive real-time data originally produced (measured or synthesized) by the platform's computer, such as pose estimates, measurements from sensors such as from laser scanners (2 LMS151 and 1 LMS200), Kinect 3D and certain encoders. Due to a combination of adequate settings for compression and sample rates of the massive measurements, all the data is well managed by a 1.5Mbps communication link.

The full availability of "raw" data (for processes that run on the remote nodes) allows flexible and sophisticated processing. The laser 3D imagery is re-synthesized on the remote nodes in place of being shared, as it is more efficient to share the original raw data (slightly degraded by lossy compression and then performing the data fusion again) than sharing the processed data (3D imagery). This policy is justified by the fact that communication resources were scarcer than processing resources. The extra processing needed for re-generating the 3D imagery is low in comparison to the extra bandwidth that would be necessary for sharing the full 3D images themselves.

The massive measurements produced by laser scanners and Kinect Cameras are compressed by lossy approaches before

Authorized licensed use limited to: Xi'an Jiaotong-Liverpool University. Downloaded on April 02,2022 at 06:43:42 UTC from IEEE Xplore. Restrictions apply.

being shared through the communication network. Laser scans are compressed by a PWL compressor and Kinect 3D frames are compressed by PWML approximations (as described in section VII).

A detailed description of the "Data Replication" approach was presented in [8].

VII. COMPRESSION OF 3D DATA

Compression is relevant for allowing the sharing of massive resources such as 3D images from 3D cameras (Kinect) and laser scanners. The rest of the sensors (such as IMU, encoders, GPS) do not contribute in a way that can impact in the communication resources.

2D laser scans are compressed by an approach that approximates the sequence of ranges in each scan by a PWL (Piece Wise Linear) approximation, given a desired tolerance.

3D images from Kinect cameras are compressed by a 2D equivalent of the PWL called PWML (Piece Wise Multi Linear). Details of those approaches and other hybrid approaches are presented in previous work ([8],[9]).

Each 3D frame can be understood as a discrete 2D function (of its "polar" parameters, azimuth and elevation angles), as an image where the pixels are represented by a 1D property (depth).

The PWML approximation for a 2D function is defined as follows,

$$\begin{gathered} \left\| \hat{f}(u,v) - f(u,v) \right\| < \tau \\ \forall (u,v) \in \Omega \\ \hat{f}(u,v) = a_k \cdot u \cdot v + b_k \cdot u + c_k \cdot v + d_k \\ \forall (u,v) \in \Omega_k \\ \{\Omega_k\}_{k=1}^{N} \quad / \quad \bigcup_{k=1}^{N} \Omega_k = \Omega \end{gathered} \tag{8}$$

Where the function $f(u,v)$, evaluated in a domain Ω, can be approximated by a Piece Wise Multi Linear function $\hat{f}(u,v)$. Its domain, Ω, can be partitioned into rectangular regions $\{\Omega_k\}_{k=1}^{N} \quad / \quad \bigcup_{k=1}^{N} \Omega_k = \Omega$. In each region the function $\hat{f}(u,v)$ is implemented by a MultiLinear expression, $\hat{f}(u,v) = a_k \cdot u \cdot v + b_k \cdot u + c_k \cdot v + d_k$, that is defined by the set of parameters $[a_k, b_k, c_k, d_k]$ associated to the region Ω_k. By defining a distance function, e.g. $E = \max_{(u,v)\in\Omega} \left(f(u,v) - \hat{f}(u,v) \right)$, an optimal approximating function $\hat{f}(.,.)$ can be found. The optimality criterion focuses on minimizing the "size" of the approximating function, i.e. the amount of data needed for storing the representation. The "size" of the approximating representation is proportional to the number of regions, ***N***. The free parameter in this optimization process is the partitioning set $\{\Omega_k\}_{k=1}^{N}$

The optimization problem can be simplified if the selection of regions is constrained to be defined by a Quad-Tree approach and the regions being square. This new constraint is introduced in order to make the optimization process real-time feasible. Consequently the solution is suboptimal but highly efficient.

A typical result of this approximation process can be seen in Fig. 13. The original surface was sampled as shown in Fig. 12, generated from a Depth image (Fig. 11) provided by a Kinect camera. If the Depth image is approximated by a PWML representation and then projected to 3D, a piece wise approximation of the 3D surface is then obtained. The PWML surface that is shown in Fig. 13 is a compressed version of the point cloud shown in Fig. 12.

A by-product of this compression process is that the PWML representation can be easily used for classifying the terrain (as presented in [9]), which is necessary for planning purposes.

In the surface classification process each PWML patch can be classified according to is position, inclination and size. This fact is exploited as expressed in the next section.

The approximation can be used on diverse types of 3D frames, from the Kinect camera or generated by the 3D laser scanner, as in both cases the 3D images can be parameterized as 2D depth images.

VIII. TERRAIN CLASSIFICATION

Terrain classification is necessary for the operation of the platform itself although it is a useful by-product as well.

The objective of the process in charge of classifying the terrain can be divided in two. Firstly a short term representation of the context is necessary for the safe operation of the platform that needs to be aware of imminent risks such as obstacles, potholes, etc. Secondly, the platform needs to plan its long term path based on a cost-to-go function. The estimation of a cost-to-go function is intended to quantify how expensive it is to reach a specified destination.

The cost of travelling is a function of diverse range characteristics of the terrain, such as slope and smoothness of the terrain.

Each of these properties can define a 2D function *F(x,y)* that can be approximated in a discrete fashion through an "Occupancy Grid".

Fig. 15 shows a map that represents such a property. The dark cells indicate that the terrain is not traversable, e.g. due to the existence of wall, pothole or other class of obstacle. Light gray indicate that those cells are unknown (i.e. no information has been collected from there, consequently the property value has not been inferred at those points). Dark gray indicates that the terrain is horizontal and smooth (traversable).

Authorized licensed use limited to: Xi'an Jiaotong-Liverpool University. Downloaded on April 02,2022 at 06:43:42 UTC from IEEE Xplore. Restrictions apply.

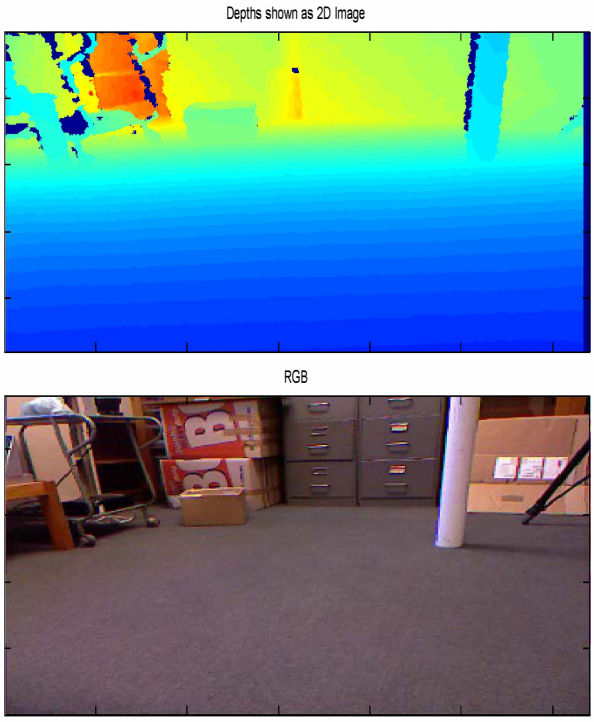

Figure 11. Depth and RGB image acquired by the on-board Kinect sensor. The image corresponds to a typical indoor context.

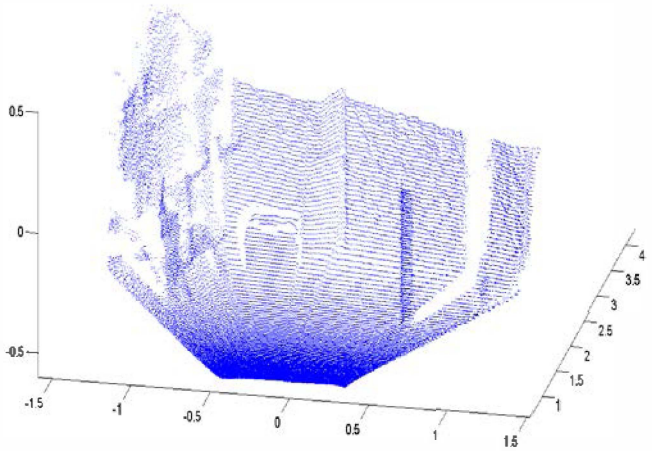

Figure 12. Points cloud for the Depth image shown in Fig. 11 (The points shown in the image is a subset of the acquired points, in order to imporve the quality of the figure).

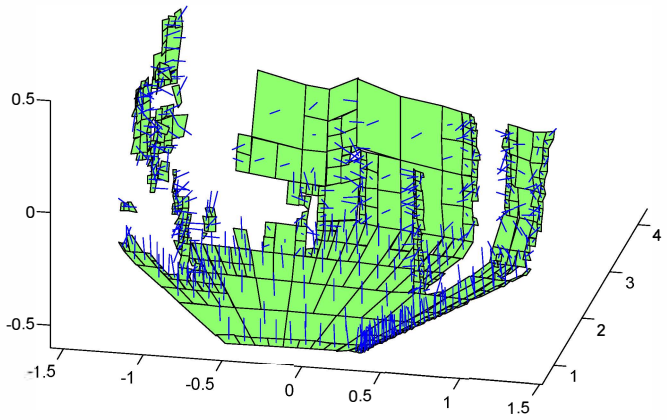

Figure 13. 3D image approximated by the PWML compressor. Each multilinear patch is classified based its normal vector (blue arrorws). In this image the ground level can be seen as modelled by a set of PWML patches that are horizontal (normal vector vertical) and at altitude close to -1 meter.

A 3D representation is a rich source of information for inferring certain characteristics of the terrain. In this work the process of generating a 2D map that describes the terrain's traversability is based on the 3D imagery. Each 3D frame is processed in order classify the terrain by analyzing the inclination and position of the surface patches. An efficient method of performing this processing is based on the PWML approximation that was described in the previous section.

The approximation naturally partitions the surface into individual patches where the surface can be well approximated by a multilinear function. Each PWML patch is then easily classified based on its geometrical properties, such as normal vectors, position and size.

Based on these properties a 2D map, where the contribution of all the PWML patches are projected, is maintained, as shown in Fig. 13.

Fig. 11 shows an image acquired by the on board Kinect camera. Both components are included: RGB and Depth information. Based on the Depth image a 3D point cloud can be evaluated as shown in Fig. 12. A PWML compressed version of the surface is shown in Fig. 13. The patches can be seen and also their associated normal vectors. A higher-level representation can be achieved by labeling the patches according to their geometrical properties (such as inclination and position). Fig. 14 shows how the patches were differently classified as based on whether or not they are traversable for the platform. Finally those patches can be located in a common coordinate frame and then processed for updating a 2D Occupancy Grid, as shown in Fig. 15.

Authorized licensed use limited to: Xi'an Jiaotong-Liverpool University. Downloaded on April 02,2022 at 06:43:42 UTC from IEEE Xplore. Restrictions apply.

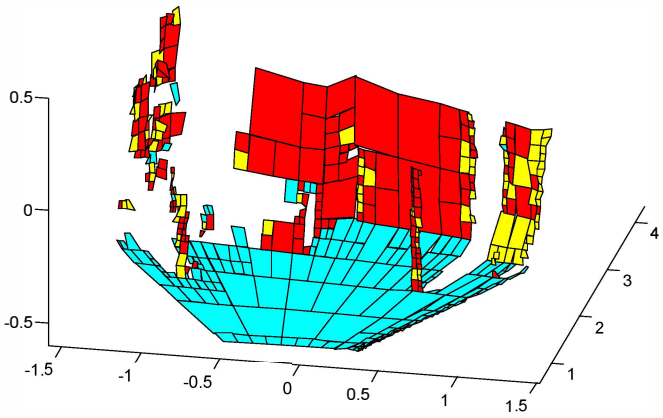

Figure 14. Classification of terrain based on the set of PWML patches. Light blue are patches inferred to be flat horizontal (traversable for the UGV). Red and yellow regions are labeled as not traversable.

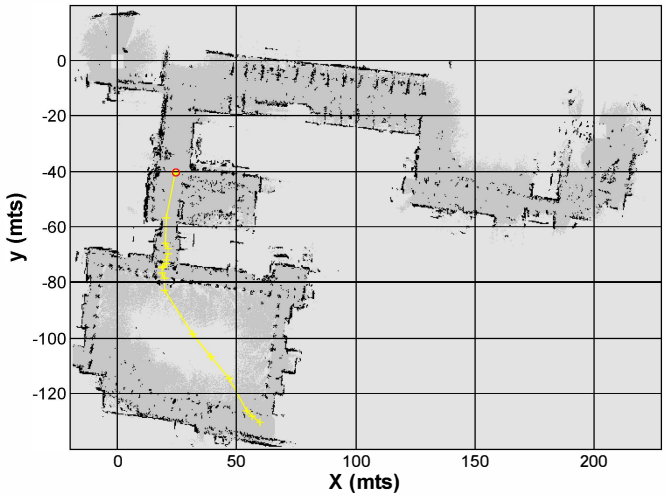

Figure 15. Occupancy Grid being generated by the processing on the remote Base Station. The 3D map and the OG are based on the information sent by the UGV. Position if the UGV is indicated at (x=24.5m,y=-40.5m) and the currently specified destination at (x=59.9m,y=-130.4m). The currently proposed path is indicated by the yellow line. Cells shown in black were inferred to be a risk (e.g. obstacle or terrain depression), cells indicated in gray are estimated to be safe (flat traversable terrain). The rest of the cells (light gray) are considered unknown (still not explored)

IX. 3D Scan Matching

In order to improve the localization in 3D contexts (i.e. where the platform does not usually operate in a 2D flat surface) a scan matching process can be used in order to estimate the 3D position and attitude of the platform. Scan matching approaches are widely used in the robotics community for mapping and localization purposes, with the Iterative Closest Point (ICP) [13] algorithm (and its variants) the most used. A more recent approach, the Normal Distributions Transform (NDT) [14], has demonstrated to be an efficient alternative to ICP. The performance of the scan matching process for both algorithms has been well investigated in [15].

In the system discussed in this paper a variation of the NDT approach is used. Details about this version of NDT can be found in [16].

The scan matching process was then successfully used in order to localize the platform in different contexts, in both indoor and outdoor environments. In Fig. 16 an indoor context is mapped by processing a sequence of the 3D images generated by the on board scanning system. Two maps are shown in the figure where one map is generated by projecting the local 3D frames based on the 3D pose estimates provided by a dead reckoning process; performing the scan matching process generates a second map. This process provides two useful results: the 3D map and the localization of the platform. It can be seen that the scan matching process outperforms the dead-reckoning estimates, although those also present good short-term accuracy.

A video of this incremental process, as the platform acquires the 3D frames and those are processed by the client process that performs the 3D scan matching, is available in [17].

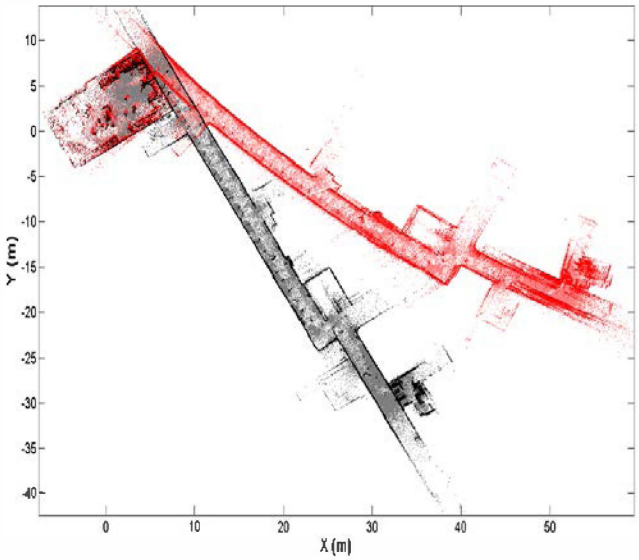

Figure 16. Part of a 3D map (generated for a period of time) by a scan matching process (in black) and for the dead reconing based one (in red). Only 10% of the points are shown in this image for the sake of simplifying it. The 3D image is shown from the top, in order to clearly see the distortion due to heading errors in the dead reckoning and the quality of the scan matching process.

X. Planning

In addition to the generated occupancy grid, Fig. 15 also shows the result of the planning process. The platform was requested to reach a destination point and to generate the 3D map of the environment. As the platform updated its belief about the context of operation it also re-planed its path in order to reach the specified destination point considering the constraints imposed by the context.

Authorized licensed use limited to: Xi'an Jiaotong-Liverpool University. Downloaded on April 02,2022 at 06:43:42 UTC from IEEE Xplore. Restrictions apply.

The planner is an implementation of the PPQ-Dijkstra (Pseudo Priority Queue) optimizer. The approach is described in [12].

XI. CONCLUSIONS

This paper presented a brief description about a system that integrates a number of fusion processes for the autonomous 3D surveying of unknown contexts. The system has the capability of planning according to the terrain characteristics, making its operation simple for the remote operators. The paper also offers a set of rich datasets that can be useful for researchers that have interest in reproducing results and testing new algorithms in the areas of perception and surveying.

XII. AVAILABLE DATASETS

Typical datasets and example source code (in Matlab) are publically available in [17].

ACKNOWLEDGMENTS

Many thanks to the following colleagues that have contributed to key parts of the UGV system: Professor Jay Kataputiya, Mr. Steve Cossell, Mr. Jim Sanderson, Mr. Stephen Kulhe, Mr. Mark Whitty and Mr. Alfred Hu.

REFERENCES

[1] Dissanayake, G.; Sukkarieh, S.; Nebot, E. and Durrant-Whyte, H.; "The aiding of a low-cost strap down inertial measurement unit using vehicle model constraints for land vehicle applications". IEEE Transactions on Robotics, Oct 2001, Volume:17, Issue:5 , pp 731-747

[2] Thrun, S., Burgard, W. and Fox, D. (2005). "Probabilistic Robotics". Cambridge, Mass: MIT Press. ISBN 0-262-20162-3.

[3] Bailey, T.and Durrant-Whyte, H.; "Simultaneous localization and mapping (SLAM): part I" Robotics & Automation Magazine, IEEE. June 2006, Volume: 13, Issue:2, pp 99 – 110.

[4] Durrant-Whyte, H. and Bailey, T.; "Simultaneous localization and mapping (SLAM): part II" Robotics & Automation Magazine, IEEE. September 2006, Volume: 13 , Issue: 3, pp: 108 - 117.

[5] G. Dissanayake, P. Newman, H.F. Durrant-Whyte, S. Clark, and M.Csobra, "A solution to the simultaneous localisation and mapping (SLAM) problem," IEEE Trans. Robot. Automat., vol. 17, no. 3, pp. 229–241, 2001.

[6] S. Thrun, W. Bugard, and D. Fox, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping," in Proc. Int. Conf. Robot. Automat., 2000, pp. 321–328.

[7] Guivant, J. "Real Time Synthesis of 3D Images Based on Low Cost Laser Scanner on a Moving Vehicle" V Jornadas Argentinas de Robótica (JAR'08), Nov 12- Nov 14 2008, Universidad Nacional del Sur, Bahía Blanca. (lcr.uns.edu.ar/jar08/papers/paper_29.pdf)

[8] Guivant, J., Cossell, S., Whitty, M., Katupitiya, J. "Internet-based operation of autonomous robots: The role of data replication, compression, bandwidth allocation and visualization". Journal of Field Robotics, 2012, volume 22, number 12. Article in Press.

[9] Robledo, A., Cossell, S., Guivant, J. "Outdoor ride: Data fusion of a 3D Kinect camera installed in a bicycle" (2011) Proceedings of the 2011 Australasian Conference on Robotics and Automation, pp. 2264-2269, December 2011, Melbourne.

[10] Whitty, M., Cossell, S., Dang, K.S., Guivant, J., Katupitiya, J. "Autonomous navigation using a real-time 3D point cloud", (2010) Proceedings of the 2010 Australasian Conference on Robotics and Automation, December 2010, Brisbane.

[11] Elfes, A. "Using occupancy grids for mobile robot perception and navigation". Computer, 1989, 22(6), pp 46-57

[12] Robledo, A., Guivant, J. "Pseudo priority queues for real-time performance on dynamic programming processes applied to path planning" (2010) Proceedings of the 2010 Australasian Conference on Robotics and Automation, December 2010, Brisbane.

[13] Paul J. Besl and Neil D.McKay. A Method for the Registration of 3-D Shapes. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2):239--256, February 1992.

[14] Peter Biber. The Normal Distributions Transform: A New Approach to Scan Matching. In Proceedings of the International Conference on Intelligent Robots and Systems.

[15] Martin Magnusson, Andreas Nuchter, Christopher Lorken, Achim J. Lilienthal, and Joachim Hertzberg. Evaluation of 3D Registration Reliability and Speed – A Comparison of ICP and NDT. In Proceedings of the IEEE International Conference on Robotics and Automation, 2009, pages 3907--3912, May 2009.

[16] Marden, S. and Guivant, J.. "Improving the Performance of ICP for Real-Time Applications using an Approximate Nearest Neighbour Search". To appear in Proceedings of the 2012 Australasian Conference on Robotics and Automation, December 2012, New Zealand.

[17] Datasets and videos: www.possumrobot.com/Datasets/datasetsIPIN2012.htm

Authorized licensed use limited to: Xi'an Jiaotong-Liverpool University. Downloaded on April 02,2022 at 06:43:42 UTC from IEEE Xplore. Restrictions apply.