Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

SLAM and data fusion from visual landmarks and 3D planes ⋆

Ayman Zureiki [1,2] **Michel Devy** [1,2]

[1] *CNRS; LAAS; 7 avenue du Colonel Roche, F-31077 Toulouse, France*
[2] *Université de Toulouse; UPS, INSA, INP, ISAE; LAAS-CNRS : F-31077 Toulouse, France*
(e-mail: {ayman.zureiki, michel.devy}@laas.fr)

Abstract:
Structured environment can be modelled in a simplified way as a set of planar surfaces and lines. For mobile robot equipped with a 3D sensor and a camera, the incremental construction of such a model is a Simultaneous Localisation And Mapping (SLAM) problem: while exploring the environment, the robot executes motions; from each position, it acquires sensory data, extracts 3D perceptual features, and simultaneously, performs self-localisation and model update. Our robot JIDO has a 3D pivoting laser range finder, acquiring images of 3D points, and has a camera. Firstly, a segmentation algorithm of a 3D image into a set of planar faces is described: this algorithm uses a region growing strategy and the Extended Kalman Filtering to estimate the parameters of the support plane of every face. These planar faces are used as landmarks. Next, we describe how to extract 2D line landmarks by fusing data from both sensors. Our stochastic map is of heterogeneous type and contains plane and 2D line landmarks. At first, The SLAM formalism is used to build a stochastic planar map, and results on the incremental construction of such a map are presented, further on, heterogeneous map will be constructed.

1. INTRODUCTION

Execute missions in initially unknown environment is still a great challenge for an autonomous mobile robot. The robot needs a description of his environment. Maps are required for self-localization, for motion planning, etc. We find in the literature two main types of maps: Topological and metric maps, see Chatila and Laumond [1985] and Filliat and Meyer [2003]. A topological map can be seen as an abstract representation describing relations between environment areas (typically, rooms or corridors). Such maps are well adapted for route planning, the selection of the best strategy for motions between areas. Their main drawback is the absence of geometric information: thus, motions are executed by sensory-motor commands (following a wall, a line, etc.). On the contrary, a metric map provides a (detailed) geometric representation of the environment; it gives explicit metric information (lengths, widths, positions, etc.), generally expressed with respect to a global reference frame.

When a robot owns a map and has to follow a given path, it executes the **Localisation** task: it estimates continuously its position in the map. The **Mapping** task is performed when a robot moves around in order to construct the map of its environment: to achieve that, the successive robot positions must be precisely known, and could be given by some external devices (GPS for example). At last, the third task, known as Simultaneous Localisation and Mapping or **SLAM**, is the conjunction of previous ones: the robot executes motions in unknown environment, and exploits relative measurements acquired by embedded sensors, to simultaneously locate itself and to build the map.

When the robot executes a SLAM task, it performs a complex process, including execution of motions, acquisition of sensory data, data association between these sensory data and the current world model, estimation of the robot pose using these associations and finally, the incremental construction of the map. It has to take into account many geometric constraints, and many sources of error. Essentially, the robustness to achieve this task depends on the robot capabilities to extract pertinent information (called Landmark) from sensory data coming from embedded sensors. The robot starts up from an initial position without any a priory knowledge about landmarks: by use of relative measurements on landmarks, the robot estimates its pose and the poses of the landmarks in an absolute frame, generally selected as the initial pose of the robot. When moving, the robot updates the landmark map and exploits it to produce an estimate of its pose.

SLAM has been an active research topic for more than twenty years; many works from Durrant-White, Tardos, Nebot, Dissanayake, Feder, Leonard, Newman, etc., aim to develop generic tools, based on the formalism of **stochastic maps** proposed by Smith et al. [1990]. The majority of these works have focused on the estimation methods required in order to maintain estimates of the robot pose and of landmark attributes in a consistent stochastic map. The extended Kalman Filter was initially proposed as a mechanism that allows the incremental fusion of information acquired by the robot; later, other methods have been exploited successfully (information filter, particle filter , etc), especially in the FastSLAM method, proposed by

⋆ A. Zureiki is supported by a PhD grant from the Syrian Government.

Thrun et al. [1998]. A well detailed state of the art can be found in Durrant-Whyte and Bailey [2006].

These approaches have been validated mainly by constructing 2D representations (like 2D segment maps) of indoor environment from laser data acquired typically by SICK range finders. Recently, 3D SLAM draws attention. Takezawa et al. [2004] describes a SLAM framework based on 3D landmarks. Jung [2004] constructs a 3D map from interest points in outer environment using stereo vision data; Sola et al. [2005] builds such maps using only monocular vision. These sparse representations allow essentially the robot to locate itself. Our work is focused on the construction of surface model in indoor environment, where many planar surfaces (ceiling, floor, walls, doors) can be used as landmarks. Our goal is to produce a geometric stochastic map made of 3D planar features. In the same area, a preliminary contribution of Nashashibi and Devy [1993], with an off line validation from a limited number of range images, and the works of Thrun et al. [2000] based on the exploitation of two laser ranger finders to acquire measurements on horizontal and vertical planes and to produce a dense model of 3D points, from which a mesh can be constructed a posteriori. Abuhadrous et al. [2004] developed a similar approach to model urban sites using GPS to localize the vehicle. Finally using only monocular vision, planes are extracted by using homographies and fused by a SLAM approach in Silveira et al. [2006].

While the algorithm of SLAM is well studied, using new sensors and robust features extraction rest an open topic. Sensors' data fusion is a interesting approach to overcome the deficiency of each sensor and to obtain more sophisticated and accurate results.

We detail in the section 2 the extraction of planar features from range images. Then in section 3 we describe our method of fusion of laser and image data in order to obtain 2D line landmarks used in the map. Next in section 4 we define our heterogeneous map which contains plane landmarks and 2D line landmarks. Finally in section 5, experimental results using our Jido mobile robot are discussed, before summarizing our contribution and presenting current works in section 6.

2. PLANE EXTRACTION

3D sensors (laser range finder, stereo vision, PMD sensor, etc.) provide images with thousands of 3D points. Compressing such a point cloud into some planar features without loosing the essential information is really important. It is a segmentation problem: how to divide the range image into features, i.e. how to bind each point with a label identifying to which feature it belongs, so that the point of the same plane have all the same label. Segmenting a range image acquired by a mobile robot, is a difficult topic, because we do not know what is seen in the scene; moreover segmentation processes must be robust in presence of non-planar or non static objects and in spite of noises.

2.1 Related Work

The planar segmentation has been well studied in computer graphics in order to perform real-time rendering of complex models Heckbert and Garland [1997]. There is a major difference between robotics and computer graphics. Data in robotics are issued from sensors and hence they are erroneous, while models in computer graphics are supposed to be without errors. The decimation algorithms in computer graphics aim to accelerate rendering and not to deal with errors.

Horn and Schmidt [1995] extract plane using Hough Transformation. They wanted to extract only vertical plane, which limit their method. Sequeira et al. [1999] use a hybrid method of *region-based* and *edge-based* to perform the segmentation and assure the alignment of consecutive data using an *Iterative Closest Point* algorithm . Liu et al. [2001] use *Expectation Maximisation (EM)* to create a 3D map of planar segments, but this iterative method is a little compatible with real-time constraint of mobile robotics. Kohlhepp et al. [2004] extract planes in real time by using an grouping algorithm of scan lines. This algorithm assemble neighbour line segments in an efficient way, but it requires data line segmentation in each scan line.

Hähnel et al. [2003] proposed a simplification algorithm adapted to robotic context. In this article they extract planes by using an approach of type *region-growing* by starting from an arbitrary point, then try to enlarge the region in all directions. Weingarten [2006] proposed some improvement to this algorithm by starting *region seed* from the most flat point in the cloud (minimum local error), and by profiting from the structure of the range image to simplify the research of neighbour points. Our approach is based on these two works, with some differences in the choice of plane's parameters and the method of their estimation. Recently, Harati et al. [2007] proposed a method based on bearing angle, which is the angle of the laser beam and the reflecting surface.

2.2 Plane Equation

In Euclidean space, a plane equation is given by:

$$a\,x + b\,y + c\,z + d = 0 \qquad (1)$$

The normal vector is $\mathbf{n} = (a\ b\ c)^t$, and the unit normal vector $\hat{u} = \frac{\mathbf{n}}{||\mathbf{n}||}$. The distance from the origin is given by $\rho = \frac{d}{||\mathbf{n}||}$. The *Hessian Normal Form* is:

$$\hat{\mathbf{u}} \cdot P + \rho = 0 \qquad (2)$$

where $P = (x\ y\ z)^t$ is a point of the plan. In these representations, there are four parameters, then there exists a redundancy, as a plane can be parametrised by only three parameters: the distance from the origin and two angles. Let φ be the angle between the projection of the plane normal on the OXY plane and the axis $\overrightarrow{OX}$, and let ψ be the angle between the plane normal with the axis $\overrightarrow{OZ}$. The plane equation is then:

$$\cos\varphi\,\sin\psi\;x + \sin\varphi\,\sin\psi\;y + \cos\psi\;z + \rho = 0 \qquad (3)$$

The vector $(\rho\ \varphi\ \psi)^t$ will be used as the minimal parametric representation of a plane.

2.3 Estimation Process

Kalman Filter is a recursive estimator: to estimate the current state, only the previous state and actual measurements are required. The observation history is not needed.

In the Extended Kalman Filter (EKF), the dynamic and observation models could be non-linear functions. To estimate the parameters of a plane using EKF, the state vector is $\mathbf{S}_t = (\rho_t \;\; \varphi_t \;\; \psi_t)^t$ and $\mathbf{P}_{t|k}$ is the covariance matrix at time t knowing all the measurements until time k. We consider that each point that belongs to the plane is an observation of this plane. we will not detail the filter equations as they can be found in many textbooks.

2.4 Segmentation by Region-growing

Let $\mathrm{V} = \{\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_\mathrm{N}\}$ be the set of 3D points, and $\mathbf{N} = \{\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_{N_\mathrm{v}}\}$ be the set of estimated normals for these points. For each point, we find the local plane (by using its 8 neighbours) using a mean least squares method, and calculate the corresponding error, let $\mathbf{E} = \{e_1, e_2, \ldots, e_{N_\mathrm{v}}\}$ be the set of these errors. In the main loop of algorithm 1, we choose a new point (not already treated) with the minimum local fitting error, we use the parameters of this local plane to initialise the state of the Kalman filter, then we call the second algorithm for *region growing*. We use a queue (First In First Out) to keep points during the (*breadth-first*) research phase. In the algorithm 2, we take the first point in the queue, then we search for its non-treated neighbours. A chain of tests are done on those neighbours: distance between the two points, distance between the neighbour point and the estimated plane, distance between the plane normal and the point normal, we do also a χ^2 test with the Mahalanobis distance. A point that satisfies all theses tests is a good candidate to joint the plane, and eventually its neighbours, hence we add it the queue, and update the filter state using the point coordinates as a new measurement. The algorithm 1 gives a pseudo-code of the segmentation process, while the algorithm 2 details the enlarging loop of the plane.

Algorithm 1 Planar Segmentation by Region-growing

Parameters:
$\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{N_\mathrm{v}}\}$: The 3D points set
$\mathbf{N} = \{\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_{N_\mathrm{v}}\}$: points' Normals
$\mathbf{E} = \{e_1, e_2, \ldots, e_{N_\mathrm{v}}\}$: errors of local planes fitting
$\mathbf{S} = \{s_1, s_2, \ldots, s_{N_\mathrm{S}}\}$: set of output planar segments
calculate normals
fit local planes
$S \leftarrow \phi$
$\mathbf{q} \leftarrow \phi$
$N_t \leftarrow 0$: number of treated points
while $N_t \leq N_\mathrm{v}$ **do**
  $\mathbf{v}_{min} \leftarrow$ get Non Treated Point With Min Local Error
  Initialise Filter using v_{min} local plane
  $\mathbf{q} \leftarrow \mathbf{v}_{min}$
  growRegion($\mathbf{q}$)
  $S \leftarrow$ add The Segment
end while

2.5 Choice of Plane Landmark Local Reference

Let $\mathcal{P}$ be a plane landmark defined by its parameters $(\rho_w, \varphi_w, \psi_w)$ in the global reference frame $\mathcal{R}_w$. We are looking for a orthonormal frame for this plane. We choose the projection of the origin O_w on the plane $\mathcal{P}$ as an origin O_p of local frame, and the axis Z_p to be parallel to the normal vector $\mathbf{n}$. We need also to choose the axis X_p. Let $\overrightarrow{i_w}, \overrightarrow{j_w}, \overrightarrow{k_w}$ be the unit vectors of axes O_wX, O_wY, O_wZ respectively, and $\overrightarrow{i_p}, \overrightarrow{j_p}, \overrightarrow{k_p}$ unit vectors of wanted axes O_pX_p, O_pY_p, O_pZ_p respectively.

Algorithm 2 Region Growing

while $\mathbf{q} \neq \phi$ **do**
  $\mathbf{v}_f \leftarrow$ getFirst($\mathbf{q}$)
  $\mathcal{N}_{\mathbf{v}_f}$: get Valid Non-Treated Neighbours
  for all $\mathbf{v}_i \in \mathcal{N}_{\mathbf{v}_f}$ **do**
    if $\mathbf{v}_i \in$ Plane **then**
      addAtTheEnd($\mathbf{q}$, $\mathbf{v}_i$)
      updateFilter($\mathbf{v}_i$)
    end if
  end for
  removeFirst($\mathbf{q}$)
end while

$$\overrightarrow{i_p} = [\, \sin\varphi_w \;\; -\cos\varphi_w \;\; 0 \,]^T \tag{4}$$

This vector can be interpreted as the unit vector of direction of the intersection line between the plane $\mathcal{P}$ and the plane $Z = 0$ (if they are not parallel).

But knowing that:

$$\overrightarrow{k_p} = \begin{bmatrix} \cos\varphi_w \sin\psi_w \\ \sin\varphi_w \sin\psi_w \\ \cos\psi_w \end{bmatrix} \tag{5}$$

The rotation matrix from global to local references is:

$$\mathbf{R}_{wp} = \begin{bmatrix} \sin\varphi_w & \cos\varphi_w \cos\psi_w & \cos\varphi_w \sin\psi_w \\ -\cos\varphi_w & \sin\varphi_w \cos\psi_w & \sin\varphi_w \sin\psi_w \\ 0 & -\sin\psi_w & \cos\psi_w \end{bmatrix} \tag{6}$$

and the translation vector is:

$$\mathbf{t}_{wp} = \rho_w \begin{bmatrix} \cos\varphi_w \sin\psi_w \\ \sin\varphi_w \sin\psi_w \\ \cos\psi_w \end{bmatrix} \tag{7}$$

3. 2D LINE LANDMARK EXTRACTION

To define a 3D line we need to define two planes. Using the camera, we can obtain one of them, so we need to use the 3D laser to define the other plane. By fusing the data of both sensors we can extract 3D lines in the scene. For representation reasons, we will consider the 3D line as 2D line attached to a holding plane. The holding plane is define by the laser data (as describe in 2).

3.1 Line Extraction

We use a traditional method of line extraction in images. It begins by a Canny filter to extract the contour, then we use a polygonal approximation to estimate the line segment passing through adjacent contour points. A phase of post processing is necessary to merge similar segments and to remove very small ones.

3.2 Interpretation Plane

For a line segment l_i in the image, the associated *Interpretation plane* is the plane passing through this 2D line and the centre of projection (viewpoint) of the camera. The normal vector of this plane can be calculated only based on intrinsic parameters of the camera $(\alpha_u, \alpha_v, u_0, v_0)$ and the data image of the segment. In fact, let (δ_i, γ_i) be the 2D line parameters of the infinite line holding the 2D segment l_i, where γ_i is the angle with the axis u and δ_i is the distance from the origin.

The 2D line equation is in the image reference frame:

$$\cos\gamma_i \;\; u + \sin\gamma_i \;\; v - \delta_i = 0 \tag{8}$$

Then using camera coordinates:

$$\cos\gamma_i \;\; (\alpha_u \frac{x_c}{z_c} + u_0) + \sin\gamma_i \;\; (\alpha_v \frac{y_c}{z_c} + v_0) - \delta_i = 0 \tag{9}$$

we obtain:

$$\begin{aligned} &\alpha_u \cos\gamma_i \;\; x_c + \alpha_v \sin\gamma_i \;\; y_c + \\ &\quad +(-\delta_i + u_0 \cos\gamma_i + v_0 \sin\gamma_i) \; z_c = 0 \end{aligned} \tag{10}$$

The normal vector in the camera reference frame is then:

$$\mathbf{n}_c = \begin{bmatrix} \alpha_u \cos\gamma_i \\ \alpha_v \sin\gamma_i \\ -\delta_i + u_0 \cos\gamma_i + v_0 \sin\gamma_i \end{bmatrix} \tag{11}$$

3.3 The 2D Line in the Plane Landmark Reference

We search to find the 2D line glued on the plane landmark that correspond to a 2D line segment in the image. Let $\mathcal{P}_i$ be a plane in the map with the parameters a_i, b_i, c_i, d_i (of course we derived them from (ρ, φ, ψ)), and l be a line segment in the image with the corresponding interpretation plane $\mathcal{P}_{sg}$ expressed in global reference frame by the parameters $a_{sg}, b_{sg}, c_{sg}, d_{sg}$. For a point (x_w, y_w, z_w) belongs to the 3D line formed by the intersection of the interpretation plane and the plane landmark, it verifies:

$$\begin{cases} a_i x_w + b_i y_w + c_i z_w + d_i &= 0 \\ a_{sg} x_w + b_{sg} y_w + c_{sg} z_w + d_{sg} &= 0 \end{cases} \tag{12}$$

Let (x_p, y_p, z_p) be the coordinates of a point in the local reference frame associated with the plane landmark $\mathcal{R}_p$. The choice of the local frame gives us:

$$\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \mathbf{R}_{wp} \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} + \mathbf{t}_{wp} \tag{13}$$

As a result, the equation of the 2D line which match with 3D line but glued on the plane landmark expressed in local coordinates (Note that $z_p = 0$):

$$\begin{aligned} &(a_{sg} \mathrm{s}_{\varphi_w} - b_{sg} \mathrm{c}_{\varphi_w}) \; x_p + \\ &\; + (a_{sg} \mathrm{c}_{\varphi_w} \mathrm{c}_{\psi_w} + b_{sg} \mathrm{s}_{\varphi_w} \mathrm{c}_{\psi_w} - c_{sg} \mathrm{s}_{\psi_w}) \; y_p + \\ &\; + (d_{sg} - a_{sg} \rho_w \mathrm{c}_{\varphi_w} \mathrm{s}_{\psi_w} - b_{sg} \rho_w \mathrm{s}_{\varphi_w} \mathrm{s}_{\psi_w} - c_{sg} \rho_w \mathrm{c}_{\psi_w}) \\ &= 0 \end{aligned} \tag{14}$$

we can write is the form:

$$\begin{aligned} &\alpha \; x_p + \beta \; y_p + \sigma = 0 \\ &where: \\ &\alpha = a_{sg} \sin\varphi_w - b_{sg} \cos\varphi_w \\ &\beta = a_{sg} \cos\varphi_w \cos\psi_w + b_{sg} \sin\varphi_w \cos\psi_w + \\ &\quad -c_{sg} \sin\psi_w \\ &\sigma = d_{sg} - a_{sg} \rho_w \cos\varphi_w \sin\psi_w + \\ &\quad -b_{sg} \rho_w \sin\varphi_w \sin\psi_w - c_{sg} \rho_w \cos\psi_w \end{aligned} \tag{15}$$

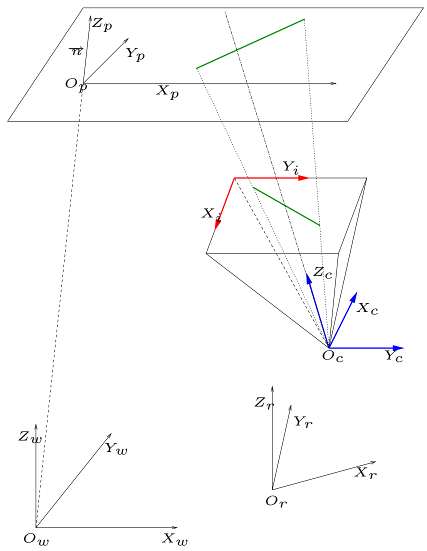

Fig. 1. A 2D Segment in image its corresponding 2D Segment glued on the Plane Landmark

4. THE STOCHASTIC MAP

The SLAM algorithm maintains a representation of Landmarks and robot states, as illustrated in figure 2. During the displacement of the robot, it uses its sensors to observe the surrounding landmarks. The system state at time k, $\mathbf{X}(k)$, is composed of the vector $\mathbf{X}_v$ representing the robot state, and of n_f vectors describing the observed landmarks, $\mathbf{X}_i(k)$, $i = 1, \ldots, n_f$.

$$\mathbf{X}(k) = \begin{bmatrix} \mathbf{X}_v^W \; \mathbf{X}_1^W \; \ldots \; \mathbf{X}_{n_f}^W \end{bmatrix}^T \tag{16}$$

where $\mathbf{X}_i^W$ is the state of a landmark in the global frame $\mathcal{R}_W$. Henceforward, (except counter indication), all states are in global frame, so we will omit the global reference symbol. We can rearrange the system state vector so that we group the states of landmarks in one term $\mathbf{X}_m(k)$:

$$\mathbf{X}(k) = \begin{bmatrix} \mathbf{X}_v \\ \mathbf{X}_m \end{bmatrix} \tag{17}$$

Our robot JIDO displaces in indoor environment supposed unknown, and composed (in a simplified way) by planar surfaces which we choose as landmarks for the SLAM algorithm. By mean of the camera we extract segments 2D in the image. These segments can be interpreted as the projection of Line 3D (or more generally Planes) onto the image plane. So the first idea to come is to use the Segment 3D as second type of landmarks in the stochastic map. We don't consider this case for many reasons: First of all, using only one camera can not give us 3D information of 3D Lines. Second, to represent 3D Lines we need at least 6 parameters (as the intersection of tow planes), but in fact, a 3D line can be represented only by 4 parameters, so by using 3D line we add non independent parameters to the map and this redundancy is a source of divergence. To over come this deficiency we choose to use the following strategy: A 3D line is the intersection of two planes, one of them must be already in the map and comes from laser data segmentation, the other is the interpretation plane of a 2D segment in the image. For representation reasons,

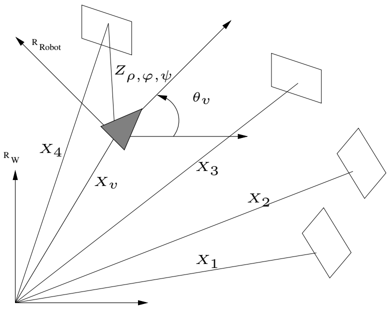

Fig. 2. System States

we choose to add the 3D line as a 2D line with respect to a local frame of a plane already in the map. Hence, the second type of Landmark for as is **a 2D Line attached to a Plane Landmark**.

This choice of 2D line landmark may appear to be correlated with the plane landmark, and does not import more information. But in reality it is not, because even if we say that the 2D line is glued with a plane, the line equation itself is derived from information from Camera and Laser. So the use of camera data adds new information not already present in laser data. We can also understand that a plane alone does not give the same information as a plane with a line on it.

In the scenario presented in figure 2. The robot state at time k can be determined by its position and orientation in the space. The robot state vector is defined by: $\mathbf{X}_v(k) = [x_v(k), y_v(k), \theta_v(k)]^T$. Each planar surface is considered as an infinite plane is defined by three parameters $\mathbf{X}_{\pi,i}(k) = [\rho_i(k), \varphi_i(k), \psi_i(k)]^T$. Each 2D Segment is considered as an infinite line in the plane landmark and is defined by means of two parameters $\mathbf{X}_{L,i}(k) = [\delta_i(k), \gamma_i(k)]^T$. Of course, a plane landmark can contain many 2D line landmarks, but a 2D line landmark can not exist alone without a holding plane landmark. Our stochastic map is then a heterogeneous map. It has two types of landmark. For more details about the construction of the stochastic map you can see Zureiki et al. [2008].

5. IMPLEMENTATION AND RESULTS

In our experiments, we used our robot JIDO illustrated in figure 3. It has two motorised wheels, a Sick Range Finder fixed on the rear and another Sick LMS-200 Range finder on a rotating axis installed ahead, a stereo rig on a pan/tilt, another stereo rig on the manipulator arm, etc.

We use the 3D *scanner* laser. It is a LMS 200 *Range Finder*, fixed on a motorised axis by stepper motor, and can rotate around the horizontal axis. The angular resolution of the laser *scanner* is fixed on $0.5°$, with a field of view of $180°$ which gives 361 points per *scan*. For the rotation of scanner around the horizontal axis, we choose to make steps of 0.01 Rad ($\approx 0.57°$) and to rotate the *scanner* between -0.3 Rad and 1.4 Rad, which includes 171 scans. The produced range image is composed of $171 * 361 = 61731$ points. We use the left camera of the stereo rig to acquire images.

The robot did a tour in our laboratory. It moves and halts, takes measurements from each position, then it advances again. It has made a tour in a corridor and return to

Fig. 3. The mobile robot Jido.

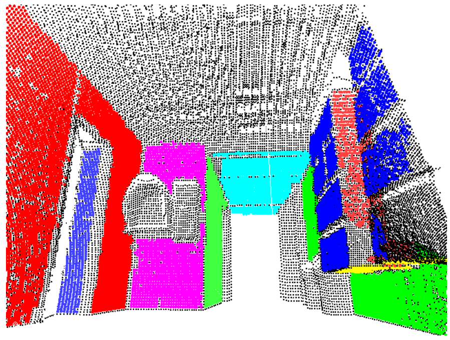

Fig. 4. Segmentation of 3D Range Image.

Fig. 5. Image of the same segmented scene.

the departure point, making in all 12 displacements. The segmentation algorithm of the range image gives good results. Figure 4 represents an output of the segmentation algorithm. In which the points of each plane are coloured by a different colour. Figure 5 shows the same scene as viewed from the camera, of course the two sensors has not the same field of view. The test is done on a P4 with 3GHz and 512MB of RAM. The segmentation takes about 10 seconds for a range image of 61731 points.

The incremental construction of the map of the corridor is illustrated (partially) in the figure 6, where we choose to print only the points belonging to each planar facet in the stochastic map, in which the poses of the robot issues from odometer are in red and from SLAM algorithm are in blue. The figure 7 represents the same scene viewed by an external camera to better appreciate the results. For now only plane landmarks are added to the stochastic map, the addition of 2D line landmarks is under construction, with the aim to present final results in future occasions.

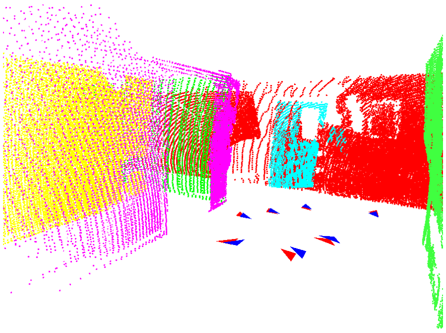

Fig. 6. A 3D Map by the SLAM algorithm.

Fig. 7. Image of the Corridor.

6. CONCLUSION

This paper has described a 3D SLAM method, proposed to build a 3D planar representation of an indoor environment from sensory data acquired by a pivoting laser range finder and a camera. A range image segmentation method is detailed to obtain planar facets from cloud points. Preliminary results on map reconstruction only with planar landmarks are presented. A 2D line landmark attached to a plane facet is proposed and extracted by fusion of laser and camera data. Future work is to achieve the building of the heterogeneous map.

Adding 2D lines to planes has two major importance: make the map more rich for navigation, and at the same time enforce the phase of data association of plane landmarks.

Currently, the robot must stop at each position during the acquisition of laser scanner. We aim to use the same segmentation method with online acquisition made by a PMD sensor (Swiss Ranger from the CSEM company) mounted on the mast of the JIDO robot.

REFERENCES

I. Abuhadrous, S. Ammoun, F. Nashashibi, F. Goulette, and C. Laurgeau. Digitizing and 3d modelling of urban environments using vehicle-borne laser scanner system. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

R. Chatila and J. P. Laumond. Position referencing and consistent world modeling for mobile robots. In *Proc. IEEE International Conference on Robotics and Automation*, 1985.

Hugh Durrant-Whyte and Tim Bailey. Simultaneous Localization and Mapping (SLAM): Part I & II. *IEEE Robotics & Automation Magazine*, 2006.

D. Filliat and J.A. Meyer. Map-based navigation in mobile robots-I. A review of localisation strategies. *Journal of Cognitive Systems Research*, 4:243–282, 2003.

D. Hähnel, W. Burgard, and S. Thrun. Learning compact 3d models of indoor and outdoor environments with a mobile robot. *Robotics and Autonomous Systems*, 2003.

A. Harati, S. Gachter, and R. Siegwart. Fast range image segmentation for indoor 3d-slam. In *The 6th IFAC Symposium on Intelligent Autonomous Vehicles*, 2007.

Paul S. Heckbert and Michael Garland. Survey of polygonal surface simplification algorithms. Technical report, Carnegie-Mellon Univ., 1997.

J. Horn and G. Schmidt. Continuous localization for long-range indoor navigation of mobile robots. In *IEEE International Conference on Robotics and Automation*, 1995.

Il Kyun Jung. *Simultaneous localization and mapping in 3D environments with stereovision.* PhD thesis, Institut National Polytechnique de Toulouse, France, 2004.

P. Kohlhepp, P. Pozzo, M. Walther, and R. Dillmann. Sequential 3d-slam for mobile action planning. In *Proc. IEEE/RSJ International Conference of Intelligent Robots and Systems (IROS)*, 2004.

Y. Liu, R. Emery, D. Chakrabarti, W. Burgard, and S. Thrun. Using em to learn 3d models of indoor environments with mobile robots. In *International Conference on Machine Learning (ICML)*, 2001.

Fawzi Nashashibi and Michel Devy. 3d incremental modeling and robot localization in a structured environment using a laser range finder. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 1993.

V. Sequeira, K. Ng, E. Wolfart, J.G.M. Goncalves, and D. Hogg. Automated reconstruction of 3d models from real environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(1):1–22, Feb. 1999.

G. Silveira, E. Malis, and P. Rives. Real-time robust detection of planar regions in a pair of images. In *Proc. IEEE/RSJ International Conference on Intelligent Robots Systems*, 2006.

R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous robot vehicles*, pages 167–193, 1990.

J. Sola, A. Monin, M. Devy, and T. Lemaire. Undelayed initialization in bearing only slam. In *Proc. IEEE/RSJ International Conference on Intelligent Robot and Systems (IROS)*, pages 2751–2756, 2005.

A. Takezawa, D. C. Herath, and G. Dissanayake. Slam in indoor environments with stereo vision. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31(1-3), 1998.

S. Thrun, D. Fox, and W. Burgard. A real-time algorithm for mobile robot mapping with application to multi robot and 3d mapping. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2000.

Jan Weingarten. *Feature-based 3D SLAM.* PhD thesis, École Polytechnique Fédérale de Lausanne, 2006.

A. Zureiki, M. Devy, and R. Chatila. Slam and multi-feature map by fusing 3d laser and camera data. In *Proc. the 5th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 2008.