The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Novel Robotic 3D Surface Mapping using Range and Vision Fusion

Blair Howarth, Jayantha Katupitiya, Jose Guivant, Andrew Szwec
School of Mechanical and Manufacturing Engineering, UNSW, Sydney, Australia

Abstract— **This paper describes a novel approach to surface fitting for the creation of a 3D surface map for use by a small articulated wall-climbing robot. Both a laser range finder and a low-resolution camera are used to acquire data in a sparse manner. By scanning at large intervals, such as every 5-10°, and then fusing the data, it is shown that it is possible to fit planar surfaces at an accuracy comparable to dense range scanning. Infinite planes are fit to lines extracted from the range scans and then the image corners and lines are used to provide polygon boundaries on these planes. This method is faster and more flexible, both in acquiring data and in computing the planar features and less memory is required. This method also works well in feature poor environments where stereo vision can struggle and does not need to process the feature correspondences in the typical fashion which also saves time. This surface fitting approach is demonstrated using a real data set and results show promise in providing quick yet accurate 3D planar surfaces which could be integrated into SLAM and motion planning frameworks.**

I. INTRODUCTION

In the field of mobile robotics one of the main research areas is that of mapping and navigation. A robot must develop some understanding of the environment it is placed in for it to be able to undertake its designated functions. Mapping is the process of building such an understanding of the immediate surroundings of a robot and is linked in with other aspects such as navigation, and localization. This combination lead to the necessary development of Simultaneous Localization and Mapping (SLAM) methods [1], where the main focus has been in the area of 2D SLAM. This is the easier problem and has been sufficient for most robotic systems. However in recent years there has been more interest in extending existing methods and developing new approaches to mapping, navigation and localization in 3D.

While many of these newer methods have shown success in 3D environments, most are mere extensions of the 2D case. Some are 3D maps built on top of 2D SLAM engines [2], whilst those implementations of full 3D SLAM are generally limited to ground constrained robots using wheels or tracks [3]. The exception to this has been in the areas of underwater or aerial vehicles [4]. In the vast majority of these 3D cases the environments of interest are large and the aim is to maintain consistent maps over large distances and long trips. In contrast there have been few attempts at true 3D SLAM for indoor robots. 2D SLAM is usually considered sufficient, with 3D mapping overlayed if needed.

Our robot (shown in Fig. 1) operates in indoor environments but is not constrained to the ground. It uses suction cups to climb walls and is able to walk on any suitable flat surface. Section II provides further details. It moves in 3D and so needs a full 3D SLAM implementation containing planar surfaces to allow this. The robot scans surfaces while it is attached to them so the ranges of interest are typically less than 500mm. As our robot walks on the surfaces rather than just mapping their location, the need for a highly accurate 3D surface mapping system becomes apparent.

Our robot is small and payload limited. All hardware must be supported by the suction cups and thus sensing and processing hardware is limited. This leads to a secondary consideration that the algorithm be lightweight and aimed at operating on a minimum of computational resources. This would allow our method to be generalized for use in other miniature robots with similar constraints.

This paper describes a method for mapping suitable planar surfaces to be later used in the aforementioned situation. Section II describes the target robot and sensor system. An overview of our mapping approach is given in Section III with details of the plane fitting and polygon bounding in Sections IV and V. The results of this approach are shown in Section VI.

II. HARDWARE

A. Robot Hardware

The target robot for our 3D mapping work is a small custom built wall-crawling robot, shown in Fig. 1. It is currently under construction and is based on our first prototype robot, shown in Fig. 2, which did not possess any sensing hardware. The new version has two feet, each with a suction cup for attaching to walls or any other suitable flat surface. The robot is a 7 degree of freedom (DOF), entirely self contained, bipedal robot. It has been designed to maximize the ability to access confined spaces. It can negotiate environments such as air ducts, pipelines and industrial plant with inclined and vertical surfaces.

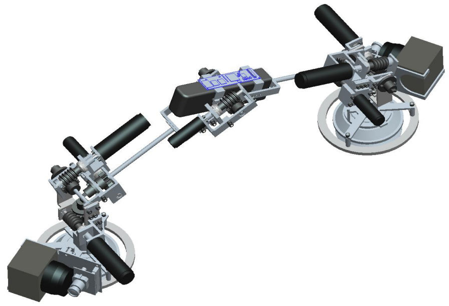

Fig. 1. Climbing Robot Mk2 Model - Target System.

The robot carries two low resolution cameras and two laser range finders (LRF) mounted in pairs, one camera and one LRF per foot. During operation the robot secures itself via one foot, while the other foot is used to make observations of the environment. The free foot, with the camera and LRF mounted, can rotate to achieve any pose and orientation within its workspace to scan the environment. The final joint of the robot rotates to tilt the sensors through an angle θ and provide a 3D scan. This action is demonstrated in Fig. 3.

B. Sensors

The chosen sensors must be small and light weight and so a Hokuyo URG04-LX LRF and a CMUCam3 camera were chosen. The LRF has a range of 4m with a resolution of 1mm and a quoted accuracy of ±10mm. It scans in 2D through 240° in steps of 0.36°. The CMUCam3 is a low-resolution CMOS RGB camera module. The resolution used was 176x287 pixels.

The Laser Range Finder was calibrated to determine the centering offset. The camera was calibrated to determine the intrinsic parameters using a standard checkerboard pattern [5]. The parameters found were the Focal Length (f_u,f_v) and the Principal Point (u_0,v_0) which form the standard camera intrinsic parameter matrix K. The extrinsic calibration between laser and camera coordinate frames was performed manually.

III. MAPPING METHOD OVERVIEW

This section describes our approach to the fitting of planar surfaces. These surfaces can be used later to create a 3D map of a robot's immediate environment or they can be integrated within a SLAM framework. The general approach taken in the literature involves dense range images from either a 3D laser range finder [6][7] or a stereo vision system [8][9]. Their range images were segmented through either region growing or edge based methods and then planes were fit to each segmented point cloud. This requires dense scanning and the segmentation can be expensive and slow.

Our proposed approach differs in that the fusion of sparse laser and image data is used to generate the planar surfaces from a minimal number of scans. While the idea of using both range and vision information to produce better 3D maps is not a new one, our approach is novel. The simplest fusion approach involves applying texture to the 3D map from the range data, but this is purely for visualization. Both [10] and [11] used vision to interpolate between range scans. They used dense scanning and high resolution images to provide accurate and detailed models but were computationally expensive and only produced point clouds. Vision has been also used to aid in segmentation [12] and to add additional range points to occluded areas in the laser range image [8][13].

A. Approach

In our method, at each pose a series of scans are acquired by tilting the sensor suite. Lines are extracted from the 2D laser range scans and these are used to fit infinite planes. Corners and lines are extracted from the camera image and these are used as polygon boundaries on the infinite plane. This allows for very sparse and fast scanning of the environment as very few scans are needed to map the major planes in the scene. Scans could be taken every $5° - 10°$ as opposed to other methods requiring a tilt angular resolution of about $1°$. The only similar idea was hinted at by [14] however their system was primarily aimed at motion estimation in large outdoor environments and their 3D reconstruction shows no resemblance to our work. Our method also aims to minimize the memory and processing requirements. This is achieved through the low frequency of scans and other data reduction techniques and approximations.

B. Feature Data Association

The foundation of our mapping approach are the features extracted from the laser and camera data to which the surfaces are fit. Once these features have been extracted, a method is needed for grouping together the features that appear to have been generated by the same surface. This is an important step to ensure that the plane fitting and bounding processes are accurate. This feature grouping is a data association problem and is unique to our approach, as opposed to other standard methods in the literature.

Each laser scan is segmented to extract the linear segments. Segments generated by a particular plane will be coplanar, taking into account the noise and uncertainties in the system and sensor. Two lines are coplanar if they intersect, meaning the shortest distance between them is less than some threshold defined by the noise. This general method involves matching segment pairs from different laser

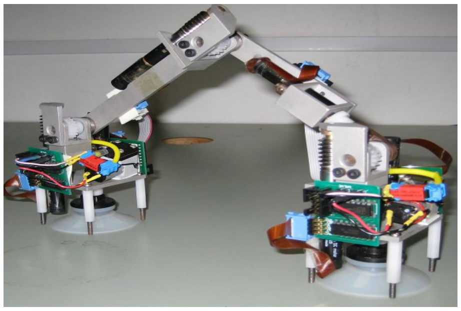

Fig. 2. Climbing Robot Mk1 - Original Prototype.

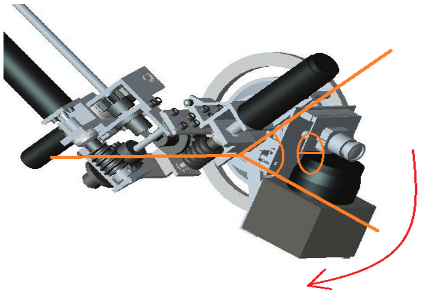

Fig. 3. Scanning Motion for Data Acquisition.

scans. However this will not work with the simplified scanning method used as it produces parallel scan lines. Instead the line segments are mapped onto the camera images and segments that are physically close and have similar texture are used to produce the initial candidate plane groupings.

The image corner and line features are pre filtered to remove false features generated by texture that do not correspond to physical edges. The remaining features are then matched to the plane groups by using the laser line segments on each image as seed regions. The image features surrounding each segment are matched to that segment providing they meet certain conditions such as similarity of texture/colour. This could be considered to be similar to a region growing approach but using a set of sparse features.

IV. PLANE FITTING

Plane fitting is the first half of the surface mapping approach. The input laser range data is segmented into straight lines in a 2D line segmentation process [15]. Image edges can also be used to complement this segmentation process by projecting the line segments on each image. This can be used for either pre splitting or post pruning of the segments to provide more robust segmentation. Each linear segment consists of a set of points that support that line. After taking at least two laser scans at different tilt angles θ, it is possible to start fitting planes. It is assumed that the surface generating the line segments is planar. While this will not always be the case, by using only good line segments a number of possible non-planar surfaces will be immediately ignored. A third line segment from further scanning will verify the planar assumption as it should be coplanar.

For each line segment, the supporting points are transformed from the local laser coordinate frame into 3D points $\mathbf{p_w}$ (with covariance Σ_w) in the world coordinate frame using the kinematics of Fig. 3. A plane is fit to the set of points from at least 2 such matched segments using Principal Components Analysis (PCA) using the plane model in (1). The parameters are a normal vector $\mathbf{n}$ which is perpendicular to the plane and d which is the orthogonal distance to the origin, ie the distance along the normal vector direction.

$$\mathbf{n} \cdot \mathbf{p} - d = 0 \tag{1}$$

PCA generates an orthonormal basis for a set of data points such that the greatest variance lies in the direction of the first Principal Component (PC), the second PC then accounts for as much of the remaining variance as possible and so on. PCA can be used to provide an orthogonal plane fit, with the normal vector $\mathbf{n}$ being the third Principal Component vector found. This should have by far the least variance, assuming the data was generated by a plane. PCA involves finding the eigenvectors of the covariance Σ_P of some data P, in the order of highest to lowest eigenvalues.

After using PCA to find $\mathbf{n}$, the only remaining plane parameter required is d. This is found by substituting $\mathbf{n}$ and $\mathbf{p_{cog}}$ into (1) and rearranging as shown in (2). The centre of gravity point $\mathbf{p_{cog}}$ lies on the plane [6] and is the mean of the data points.

$$d = \mathbf{n} \cdot \mathbf{p_{cog}} \qquad \mathbf{p_{cog}} = \frac{\sum_i^m \mathbf{p_i}}{m} \tag{2}$$

The three principal components produced form an orthonormal basis of a new coordinate frame aligned to the fitted plane. It can therefore be used as a rotation matrix R to transform the input points $\mathbf{p_w}$ into the plane coordinate frame to give points $\mathbf{p_p}$ as in (3).

$$\mathbf{p_p} = R[\mathbf{p_w} - \mathbf{p_{cog}}] \qquad \Sigma_p = R\Sigma_w R^T \tag{3}$$

A. Uncertainty Analysis

In order to find the covariance of the plane parameters in (1), an ordinary least squares fit is applied using (4-5). It is possible to derive the covariance of the least square parameters $\beta_0, \beta_1, \beta_2$ and then use these to calculate the covariances of $\mathbf{n}$ and d. The least square fit shown below is not orthogonal and thus is not identical to the PCA fit, nor is it an optimal fit. It is however possible to first transform the data points into a new coordinate frame defined by the PCA orthonormal basis whose origin is at the centre of gravity point. In this new coordinate frame the least square fit aligns with the PCA fit. The desired covariance can then be found, converted to $\mathbf{n}$,d coordinates and then transformed back into the original coordinate frame. This approach is an extension of that used in [6].

The least squares regression equation for plane fitting can be defined as (4).

$$A\mathbf{x} + B\mathbf{y} + C\mathbf{z} + D = 0 \tag{4}$$

By rearranging as a function for $\mathbf{z}$ and changing the parameters to $\beta_0, \beta_1, \beta_2$, an amenable form is achieved as in (5).

$$\begin{gathered} \mathbf{z} = \beta_0\mathbf{x} + \beta_1\mathbf{y} + \beta_2 \\ \beta_0 = \frac{A}{-C} \quad \beta_1 = \frac{B}{-C} \quad \beta_2 = \frac{D}{-C} \end{gathered} \tag{5}$$

With the plane equation in this form, the $\mathbf{x}, \mathbf{y}, \mathbf{z}$ values are substituted for the coordinates of each point $\mathbf{p_w}$ and the equation is converted into matrix form consisting of N rows, one for each of the N points. This can be simplified and rearranged for β as in (6), where M is a matrix containing N rows of the x,y values.

$$Z = M\beta \qquad \beta = [MM^T]^{-1}M^T Z \tag{6}$$

The covariance of the plane parameters $\beta = [\beta_0 \; \beta_1 \; \beta_2]^T$ can be calculated from the input covariances of each input point. This is done using (7) where Σ is a matrix with the input covariance matrices of each point along its diagonal. F is the jacobian of the three β parameters with respect to each input point's x,y,z coordinates.

$$\Sigma_\beta = F\Sigma F^T \tag{7}$$

The jacobian F can be found by taking the partial derivatives of (6). This requires the inverse to be found symbolically rather than numerically. It is now possible to explicitly write equations for $\beta_0, \beta_1, \beta_2$ in terms of the input points x_i, y_i, z_i. The partial derivatives required for the Jacobian F can then be found and the covariance calculated, which can used to find the covariance of $\mathbf{n}$,d.

B. Covariance transformation

The input data points are transformed into the Coordinate Frame defined by the PCA axes with $\mathbf{p_{cog}}$ being the origin using (3). In this new coordinate frame, the plane will lie along the X, Y axes as expected because this coordinate frame is the plane itself. Therefore the plane parameters are exactly $\mathbf{n} = [0\ 0\ 1]^T$ and $d = 0$ but these are not of interest as PCA has already provided these values in the original coordinate frame. For the same reasons the parameters β from (5) also all equal 0. But it is possible to calculate the covariance matrix Σ_β as outlined above and use this to determine the covariance matrix Σ_n and variance σ_d.

As the data points are centred about the mean $\mathbf{p_{cog}}$ in the plane coordinate frame, many of the sum terms in the partial derivatives for β equate exactly to 0 thus simplifying the equations considerably. So by finding the partial derivatives of β, the jacobian components F_i can be found as in (8).

$$F_i = \begin{bmatrix} \frac{\beta_0}{x_i} & \frac{\beta_0}{y_i} & \frac{\beta_0}{z_i} \\ \frac{\beta_1}{x_i} & \frac{\beta_1}{y_i} & \frac{\beta_1}{z_i} \\ \frac{\beta_2}{x_i} & \frac{\beta_2}{y_i} & \frac{\beta_2}{z_i} \end{bmatrix} = \begin{bmatrix} \frac{z_i}{\sum_i x_i^2} & 0 & \frac{x_i}{\sum_i x_i^2} \\ 0 & \frac{z_i}{\sum_i y_i^2} & \frac{y_i}{\sum_i y_i^2} \\ 0 & 0 & \frac{1}{N} \end{bmatrix} \tag{8}$$

Now that the covariance matrix Σ_β has been found it can be used to find the covariance matrix Σ_n and variance σ_d, still in the plane coordinate frame. The transformation from β to $\mathbf{n}$ and d is derived and their partial derivatives are used to transform the uncertainty. Using (5) with $C = -1$, the general plane parameters are A, B, C, D. The normal vector $\mathbf{n}$ is found by normalising the vector $[A\ B\ C]^T$ and d is found by normalising D using the same factor. The jacobian of the transformation is used to find the covariance matrix Σ_n as in (9) and likewise the variance σ_d^2 is shown in (10).

$$\Sigma_n = F_n \Sigma_\beta F_n^T = \begin{bmatrix} \sigma_{\beta_0}^2 & \sigma_{\beta_{01}}^2 & 0 \\ \sigma_{\beta_{01}}^2 & \sigma_{\beta_1}^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{9}$$

$$\sigma_d^2 = F_d \Sigma_\beta F_d^T = \sigma_{\beta_2}^2 \tag{10}$$

These are the sought after uncertainty information for the plane parameters. All that remains is to transform them back into the original coordinate frame using (3).

V. PLANE POLYGON BOUNDING

Unlike other mapping approaches, the sparse nature of the range scanning does not supply enough information to determine the polygon boundaries of the infinite planes fit in Section IV. Instead our method uses image features extracted from the cameras to estimate them. The extracted image corners act as the primary basis for the boundary of the plane and the image lines provide information as to the connectedness of the corners. A modified version of [16] was used, based on canny edge detection to detect lines and then curvature to detect corners.

The simplest plane fit involves two laser lines and one camera image. In this case the polygon is bounded by the extracted corners. However by adding a second or third image it is possible to improve the accuracy of the boundary and also expand the boundary if the camera field of view is limited/occluded. Corners from new images need to be incrementally matched and fused with the current boundary corner estimates.

A. Camera CF to World CF

Prior to matching and fusing corners, the camera pixel location $\mathbf{p_i}$=[u v 1] must be transformed into world coordinates. This is only possible up to a scale factor Z.

The first step is to convert from image coordinates to normalized camera coordinates $\mathbf{p_{cn}}$ using (11). The uncertainty $\Sigma_{p_{cn}}$ is also calculated.

$$\mathbf{p_{cn}} = \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix} = \begin{bmatrix} x_c/z_c \\ y_c/z_c \\ 1 \end{bmatrix} = K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K^{-1}\mathbf{p_i} \tag{11}$$

It is then required to convert this normalized point from the camera coordinate frame to the world coordinate frame. Essentially $\mathbf{p_{cn}}$ represents a line in 3D space passing through the camera center $\mathbf{p_c}$=[0 0 0]^T and with a direction vector given by $\mathbf{p_{cn}}$. To convert this line to world coordinates, the two components need to be separately transformed. This gives the 3D line equation (12) in world coordinates with direction vector $\mathbf{v}$ and passing through $\mathbf{p_0}$ in world coordinates. The covariances Σ_v and Σ_{p_0} are also found.

$$\mathbf{p} = \mathbf{p_0} + Z\mathbf{v} \qquad Z \in \mathbb{R} \tag{12}$$

B. Line - Plane Intersection

As the converted camera pixel represents a line in 3D space, further information is needed to determine the scale factor Z and thus the unique point in 3D that was imaged by the camera. If the parameters of the surface that contains the imaged point are known then it is possible to calculate the 3D location of that intersection point. The plane parameters found in Section IV can be used by combining (12) with (1) and it is then possible to estimate the scale factor Z and thus the unique point $\mathbf{p_w}$ of the image corner using (13).

$$Z = \frac{d - \mathbf{n} \cdot \mathbf{p_0}}{\mathbf{n} \cdot \mathbf{v}} \tag{13}$$

The uncertainty of $\mathbf{v}$ and $\mathbf{p_0}$ can be propagated into Z using the Jacobian Matrix J and the input covariance matrix Σ containing the covariances Σ_v, Σ_n, σ_d and σ_{p_0} along its diagonal.

$$\sigma_z^2 = J_z \Sigma J_z^T \qquad J_z = \left[\frac{\partial Z}{\partial \mathbf{v}}\ \frac{\partial Z}{\partial \mathbf{n}}\ \frac{\partial Z}{\partial d}\frac{\partial Z}{\partial p_0}\right] \tag{14}$$

To calculate the final position of the image pixel in world coordinates the value of Z found in (13) is used to find $\mathbf{p_w}$ as in (15) and the covariance Σ_{pw} can be found using (16).

$$\mathbf{p_w} = \mathbf{p_0} + Z\mathbf{v} \tag{15}$$

$$\Sigma_{pw} = \Sigma_{p0} + \mathbf{v}\sigma_z^2\mathbf{v}^T + Z^2\Sigma_v^2 \tag{16}$$

C. Corner Fusion

New detected corners need to be matched to determine which existing boundary corner to merge with. This is done by projecting the new corner onto the current polygon boundary and using a nearest neighbor search. If the distance is smaller than a threshold, the corners are considered matched and will be merged. The fusion of new corners is based on the standard Kalman Filter equations to provide the optimal linear estimate. A motion model is not needed as the plane is stationary. The state variables $\mathbf{x_k}$ with covariance matrix P_k are the boundary corner locations on the infinite plane they have been assigned to. The measurements $\mathbf{z_k}$ are the new corner locations $\mathbf{p_w}$ with covariance Σ_w from (15-16) after having been projected onto the plane using (3).

VI. RESULTS

This approach was tested with the scene shown in Fig. 4. There are eight planar surfaces that are of interest for demonstrating our method. Seven scans were taken at tilt angles θ of 0-30° below horizontal with a step of 5°.

Infinite planes were fit to the eight surfaces as in Section IV. The resulting estimated parameters **n** and d are shown in Table I along with their known values. The plane parameter estimates for $\mathbf{n_{est}}$ show good agreement with the ground truth, however for some planes there is a large error in the d parameter. This is due to systematic LRF range errors which are not accounted for. This also leads to errors in the **n** parameter. The least squares plane fitting assumed that the range error was gaussian so the covariance of **n**,d is underestimated. Because of this bias, the polygon boundary results shown below will use the ground truth values of **n**,d. This will allow the accuracy of this approach to be demonstrated independently of this range bias. Surface 7 is not fit as only one laser scan intersected it. With the current method it was not possible to fit an infinite plane.

TABLE I

PCA Plane Fitting results.

Planar Surface	n_{est} normal vector	n_{true} normal vector	d_{est} mm	d_{true} mm
1 Wall	[-0.01 -0.02 0.99]	[0 0 1]	579	550
2 Book Front	[0.05 -0.01 0.99]	[0 0 1]	543	507
3 Foam Top	[-0.03 -0.62 0.79]	[0 -0.56 0.83]	372	353
4 Balsa Wood	[-0.03 -0.02 0.99]	[0 0 1]	576	545
5 Box Front	[-0.47 -0.01 0.88]	[-0.43 0 0.91]	203	197
6 Box Side	[-0.90 0.03 -.43]	[-0.91 0 -.43]	-290	-283
7 Box Top	-	[0 1 0]	-	80
8 Black Paper	[-0.02 0.99 -0.02]	[0 1 0]	-21	0

Surface 3 (Foam) is used to demonstrate the fusion of boundaries from multiple images. The individual boundaries are shown in Fig. 5(left) with the true boundary shown in gray. Individually each boundary may not cover the entire surface or may do so inaccurately. By merging boundary information from multiple images, both the accuracy and field of view improves to better encompass the entire boundary of the surface. The merged corners and the plane polygon boundary are shown in Fig. 5(right).

This process was repeated for all eight surfaces being tested and the resulting 3D surfaces can be seen in Fig. 6(right) as compared with an image of the scene Fig. 6(left). This image is used for comparison only as the reconstructed 3D surfaces result from the fusion of multiple images and so the perspective is not identical. It can be seen that these surfaces provide a good representation of the actual scene. Surface 7 (Box Top) is missing, as mentioned earlier. However this surface shares some image lines with surface 5 and 6 which could be used as extra lines in the plane fitting process. It should also be noted that this is only one 3D scan. The surfaces will be merged with those from other scans to improve their accuracy so it is not vital that the representation is perfect from such a single 3D scan.

For each polygon, the boundaries showed reasonable

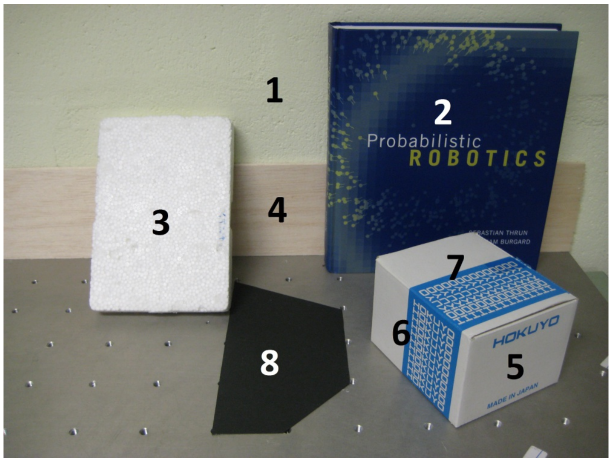

Fig. 4. Test Scene. Surfaces of interest highlighted.

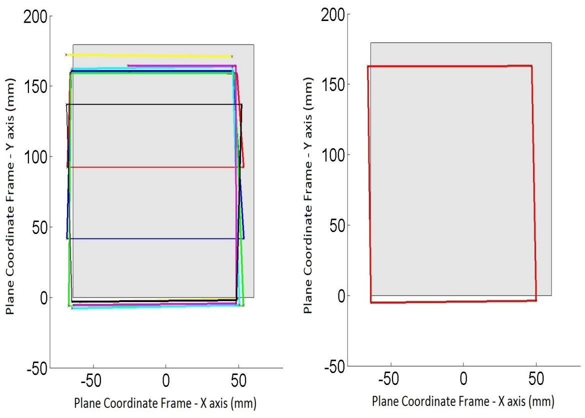

Fig. 5. Surface 3) Foam. (left) Individual polygon boundaries. (right) Merged polygon boundary. True plane boundary in shaded gray.

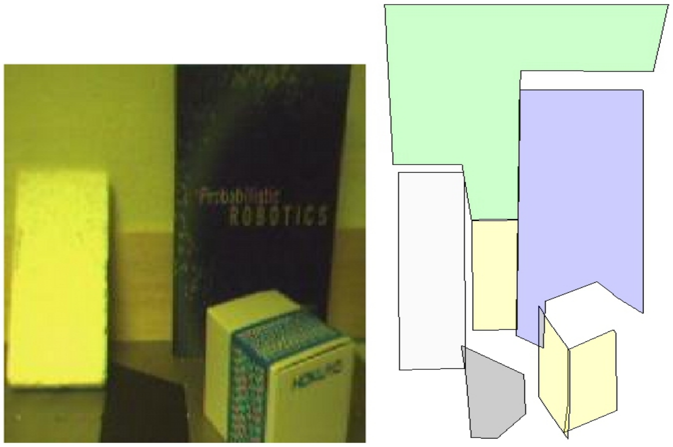

Fig. 6. 3D scene results: (left) 20° camera image. (right) 3D polygons.

agreement with the known true boundaries however some error in location is observed. To quantify this error, for each corner the error was found absolutely (mm) and as a ratio to its standard deviation. The Root Mean Square Error (RMSE) for these two error metrics is shown in Table II. While the RMSE(mm) error is reasonable, the covariances are clearly underestimated as shown by RMSE(σ).

The results show that this method is capable of producing good surface maps for each 3D scan, but with some limitations. The accuracy is significantly influenced by biases in the LRF range measurements caused by surface properties such as the material and angle of incidence of the beam. These can be hard to correct for in general. The sensor intrinsic and extrinsic calibrations are a possible cause of the bias in the polygon boundaries. These both lead to the covariances being underestimated as the plane fitting and boundary fusion assumed only gaussian errors. This could be improved by better modeling of the LRF error sources, improved calibration and by the use of non-gaussian estimators. More experiments are being developed to further test and refine this methodology on a variety of surfaces.

VII. CONCLUSIONS

This paper has successfully demonstrated a novel method for fitting planar surfaces to 3D sensor data. The approach involved the fusion of sparse range and image scans of indoor environments to fit polygon boundaries onto planar surfaces. By extracting line segments from the 2D range scans and fitting an infinite plane, it was shown to be possible to determine the points of intersection of an image corner to determine the 3D position and covariance of that image point.

TABLE II

BOUNDARY CORNER LOCATION ERROR.

Plane	RMSE (mm)	RMSE (σ)
1 Wall	-	-
2 Book Front	17.7	7.8
3 Foam Top	14.8	5.2
4 Balsa Wood	9.5	3.4
5 Box Front	15.0	2.7
6 Box Side	14.1	3.4
7 Box Top	-	-
8 Black Paper	12.3	3.3

A sample scene containing eight planes was used to demonstrate this method and it was shown to provide a good estimate of the plane boundaries. The major source of error appeared to be sensor calibration and LRF range biases. Better sensor modeling and calibration will likely provide good gains in accuracy and reduce most of the bias seen in the results. The field of view of the camera was narrow and a wider lens would provide for better boundary estimates.

This method shows promise in providing accurate yet fast 3D planar surfaces which could act as features in a 3D SLAM implementation. This would allow our climbing robot to successfully navigate its environment autonomously.

VIII. ACKNOWLEDGMENTS

This work is supported in part by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government.

REFERENCES

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (slam): Part i the essential algorithms," *Robotics and Automation Magazine*, vol. 13, pp. 99–110, 2006.

[2] I. Mahon and S. Williams, "Three-dimensional robotic mapping," *Australasian Conference on Robotics and Automation*, 2003.

[3] A. Nuchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun, "6d slam with an application in autonomous mine mapping," *Robotics and Automation, IEEE International Conference on*, vol. 2, 2004.

[4] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Autonomous Robots*, vol. 15, no. 2, pp. 111–127, 2003.

[5] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," *International Conference on Computer Vision 99*, vol. 1, pp. 666–673, 1999.

[6] J. Weingarten, G. Gruener, and R. Siegwart, "Probabilistic plane fitting in 3d and an application to robotic mapping," *Robotics and Automation, IEEE International Conference on*, vol. 1, 2004.

[7] A. Zureiki and M. Devy, "Appearance-based data association for 3d and multisensory slam in structured environment," *Information and Communication Technologies: From Theory to Applications, 3rd International Conference on*, pp. 1–6, April 2008.

[8] P. Dias, V. Sequeira, F. Vaz, and J. Goncalves, "Registration and fusion of intensity and range data for 3d modelling of real world scenes," *3-D Digital Imaging and Modeling, Fourth International Conference on*, pp. 418–425, 2003.

[9] L. Iocchi, K. Konolige, and M. Bajracharya, "Visually realistic mapping of a planar environment with stereo," *Proceesings of the 2000 International Symposium on Experimental Robotics*, 2000.

[10] H. Andreasson, R. Triebel, and A. Lilienthal, "Vision-based interpolation of 3d laser scans," *Proc. of ICARA*, pp. 469–474, 2006.

[11] J. Diebel and S. Thrun, "An application of markov random fields to range sensing," *Proceedings of Conference on Neural Information Processing Systems (NIPS), Cambridge, MA, USA*, 2005.

[12] H. Andreasson, R. Triebel, and W. Burgard, "Improving plane extraction from 3d data by fusing laser data and vision," *Intelligent Robots and Systems, International Conference on*, pp. 2656–2661, 2005.

[13] W. Jiang and J. Lu, "Panoramic 3d reconstruction by fusing color intensity and laser range data," *Robotics and Biomimetics, IEEE International Conference on*, pp. 947–953, 2006.

[14] Y. Bok, Y. Hwang, and I. Kweon, "Accurate motion estimation and high-precision 3d reconstruction by sensor fusion," *Robotics and Automation, IEEE International Conference on*, pp. 4721–4726, 2007.

[15] V. Nguyen, S. Gächter, A. Martinelli, N. Tomatis, and R. Siegwart, "A comparison of line extraction algorithms using 2d range data for indoor mobile robotics," *Autonomous Robots*, vol. 23, no. 2, pp. 97–111, 2007.

[16] X. He and N. Yung, "Corner detector based on global and local curvature properties," *Optical Engineering*, vol. 47, p. 057008, 2008.