Proceedings of the 2006 IEEE/RSJ
International Conference on Intelligent Robots and Systems
October 9 - 15, 2006, Beijing, China

Simultaneous Localisation and Mapping: A Stereo Vision Based Approach

D. C. Herath, Sarath Kodagoda and Gamini Dissanayake
ARC Centre of Excellence for Autonomous Systems (CAS)
Faculty of Engineering
University of Technology, Sydney
Broadway, NSW, Australia
Email: {d.herath, s.kodagoda, g.dissanayake}@cas.edu.au

Abstract **- With limited dynamic range and poor noise performance, cameras still pose considerable challenges in the application of range sensors in the context of robotic navigation, especially in the implementation of Simultaneous Localisation and Mapping (SLAM) with sparse features. This paper presents a combination of methods in solving the SLAM problem in a constricted indoor environment using small baseline stereo vision. Main contributions include a feature selection and tracking algorithm, a stereo noise filter, a robust feature validation algorithm and a multiple hypotheses adaptive window positioning method in 'closing the loop'. These methods take a novel approach in that information from the image processing and robotic navigation domains are used in tandem to augment each other. Experimental results including a real-time implementation in an office-like environment are also presented.**

Index Terms – SLAM, KLT, Loop closure, Stereo vision.

I. Introduction

There have been various attempts in using cameras as range sensors in robotic navigation. They in most cases fall in to two categories, namely in obstacle detection[1, 2] and in solving localization and mapping problem[3, 4]. In SLAM there is a wide and diverse range of implementations. From active stereo vision [5], static binocular and trinocular vision [6] to single camera bearing only SLAM[7]. The latter not used as a range sensor in the strict sense.

The limited dynamic ranges in current CCD imagers make them vulnerable in outdoors. In an indoor environment with more control over lighting conditions, it is possible to push further the state of the art in robotic navigation with these sensors. In the onset a feature based SLAM implementation using stereo camera looks straightforward. However several issues needed to be addressed before a robust implementation can be achieved. In this work we address some of these issues and present the results achieved with experiments conducted using a pioneer robot equipped with a stereo camera in an office environment. Especially when interpreting the results presented it is imperative to note the small baseline (≈ 0.088m) of the camera and the wide angle lenses ($\approx 90^{o}$) used. According to [8] a baseline/depth ratio of less than 1/30 would not make much sense. This translates to an inconsistent filter performance and in this exposition we present several filtering techniques that make the Extended Kalman Filter (EKF) based estimations consistent.

In this work, the well established Kanade-Lucas-Tomasi (KLT) [9] algorithm is used in tracking features between image frames essentially reinforcing the data association and a novel multiple hypotheses adaptive window positioning method is used in establishing a loop closure. Also we show with empirical evidence that an EKF based algorithm still can be inconsistent due to gross errors present in observations and limitations in the current error models that defy conventional outlier detection methods. As a solution, we present a robust data validation algorithm with substantive experimental evidence.

Rest of the paper is organized as follows. Section II contains a summary of the 3D SLAM implementation whilst a summary of the (KLT) tracking algorithm is presented in Section III. Section IV presents two methods in dealing with stereo noise. Section V describes the loop closure. In section VI we present comparative results of several SLAM implementations with our proposed implementation. Section VII concludes the paper.

II. 3D SLAM Formulation

The SLAM frame work based on Kalman filtering is well established [10], hence here we present only a summary to the 3D extension. The robot state is defined by $X_r = [x_r \quad y_r \quad \varphi_r]^T$, where x_r and y_r denotes location of the robot's rear axle centre with respect to a global coordinate frame and φ_r is the heading with reference to the x-axis of the same coordinate system. Landmarks are modeled as point features, $\mathbf{P}_i = [x_i \quad y_i \quad z_i]^T$, $i = 1,...,N$ and represented by Cartesian coordinates as in Fig. 1.

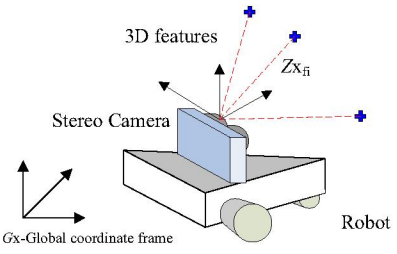

Fig. 1 The robot in 3D world coordinates observing a feature in 3d space.

A. Vehicle and landmark augmented process model

The vehicle motion through the environment is modeled as a conventional discrete time process model as in (1).

$$\begin{bmatrix} x_r(k+1) \\ y_r(k+1) \\ \varphi_r(k+1) \end{bmatrix} = \begin{bmatrix} x_r(k) + \Delta T V(k)\cos(\varphi_r(k)) \\ y_r(k) + \Delta T V(k)\sin(\varphi_r(k)) \\ \varphi_r(k) + \Delta T \omega(k) \end{bmatrix} \quad (1)$$

ΔT is the time step, V(k) is the instantaneous velocity and ω(k) is the instantaneous turn-rate. The robot is assumed to be travelling on a horizontal plane. The landmarks in the environment are assumed to be stationary point features and hence the process model is,

$$\begin{bmatrix} x_i(k+1) \\ y_i(k+1) \\ z_i(k+1) \end{bmatrix} = \begin{bmatrix} x_i(k) \\ y_i(k) \\ z_i(k) \end{bmatrix} \quad (2)$$

where, i (=1,…,N) is the landmark number. Using (1) and (2), the augmented state transition matrix for the complete system can be represented.

B. Observation model

The observation model can be represented as,

$$Z\ (k+1) = \begin{bmatrix} z_x(k+1) \\ z_y(k+1) \\ z_z(k+1) \end{bmatrix} = \begin{bmatrix} a \\ b \\ z_{fi}(k+1) \end{bmatrix} \quad (3)$$

where

$$a = (x_{fi}(k+1) - x_r(k+1))\cos(\varphi_r(k)) + (y_{fi}(k+1) - y_r(k+1))\sin(\varphi_r(k))$$
$$b = -(x_{fi}^{-}(k+1) - x_r(k+1))\sin(\varphi_r(k)) + (y_{fi}(k+1) - y_r(k+1))\cos(\varphi_r(k))$$

It is to be noted that each feature is defined by a point in 3D space, $\mathbf{x}_{fi}(k) = \begin{bmatrix} x_{fi}(k) & y_{fi}(k) & z_{fi}(k) \end{bmatrix}^T$. The measurement error covariance is,

$$\mathbf{P}_{fi} = \nabla g_{xy\varphi} \mathbf{P}_r \nabla g_{xy\varphi}^T + \nabla g_{x_{fi}y_{fi}z_{fi}} \mathbf{R} \nabla g_{x_{fi}y_{fi}z_{fi}}^T \quad (4)$$

$$\mathbf{R} = \frac{B^2}{d^2} \begin{bmatrix} \frac{f^2 \sigma_d^2}{d^2} & \frac{-uf\ \sigma_d^2}{d^2} & \frac{-vf\ \sigma_d^2}{d^2} \\ \frac{-uf\ \sigma_d^2}{d^2} & \sigma_u^2 + \frac{u^2 \sigma_d^2}{d^2} & \frac{uv\ \sigma_d^2}{d^2} \\ \frac{-vf\ \sigma_d^2}{d^2} & \frac{uv\ \sigma_d^2}{d^2} & \sigma_v^2 + \frac{v^2 \sigma_d^2}{d^2} \end{bmatrix} \quad (5)$$

where $\mathbf{P}_r$ is the error covariance matrix of the robot location estimate extracted from the state covariance matrix $\mathbf{P}(k/k)$ and $\mathbf{R}$ is the measurement noise covariance. ∇g is the Jacobean of the observation function. And $\sigma_u, \sigma_v, \sigma_d$ represents the pixel uncertainties in image u,v location and the disparity d respectively. f and B are the camera focal length and base line. Having defined the process model and the observation model, the standard Kalman filter based realization [10] is carried out.

III. KLT Implementation

Data association is very crucial in a robust SLAM implementation. One strong advantage of using vision is its ability to track features in the image plane, and hence enhancing data association. Kanade, Lucas and Tomasi [11, 12] have proposed a feature selection and robust tracking algorithm (KLT). The main advantage of KLT over other descriptor based (e.g. [4])methods is its ability to efficiently track features between consecutive images in real time. A description of the KLT algorithm is given in [9] and the complete derivation of the algorithm is presented in the unpublished note by Birchfield [13]. The tracker and the feature selection method compliment each other in that the combination is optimal by design. In this work we use KLT to extract reliable features and track them efficiently between images. The tracking algorithm is described in brief below.

Given two images I,J assuming small inter-frame displacements, image motion can be described by suitably moving every point in the current frame to achieve the next frame

$$I(x, y, t+\tau) = I(x - \xi(x, y, t, \tau), y - \eta(x, y, t, \tau)) \quad (6)$$

displacement of a point at x is given by $\delta = (\xi, \eta)$ and using the affine motion model

$$\boldsymbol{\delta} = \mathbf{D}\mathbf{x} + \mathbf{d} \quad (7)$$

where, $\mathbf{D} = \begin{bmatrix} d_{xx} & d_{xy} \\ d_{yx} & d_{yy} \end{bmatrix}$ is the deformation matrix and $\mathbf{d}$ is the displacement vector. A point at x in the first image I moves to point $\mathbf{A}\mathbf{x} + \mathbf{d}$ in the second image J.

$$J(\mathbf{A}\mathbf{x} + \mathbf{d}) = I(\mathbf{x}) \quad (8)$$

Where, $\mathbf{A} = \mathbf{1} + \mathbf{D}$ .In order to find the above motion parameters $\mathbf{A}$ and $\mathbf{D}$ minimise dissimilarity,

$$\varepsilon = \iint_w [J(\mathbf{A}\mathbf{x} + \mathbf{d}) - I(\mathbf{x})]^2 w(\mathbf{x})\, d\mathbf{x} \quad (9)$$

where $w(\mathbf{x})$ is a weighting function, which is set to 1 in our implementation. In order for the SLAM to perform real-time the number of features intialised/tracked per image was set to 20.

IV. Managing Noise

A. Noise Filter

KLT provides with good features to track. However, it can pick up illusive features in the image plane like an intersection of two far apart real world features. Those features are catastrophic for SLAM as they violate the

fundamental fixed landmark assumption. Such features can be detected by analyzing depth (disparity) discontinuities in a small support region surrounding the point of interest as follows.

Given the stereo images, a corresponding depth map (disparity image) of the scene is created. Then a small patch of the disparity image around the corresponding feature location selected by KLT is obtained (Our experiments showed that a 3x3 patch yields better results). A histogram for this small patch is generated and the mean and the standard deviation are calculated. Then the resulting standard deviation σ is compared against a predetermined threshold (0.20m). If the test is successful the feature is used in the filter. For successful new features a delayed initialization is carried out in order to assert the stability of the feature (Fig. 2.)

B. *Feature Validation Algorithm*

One of the most difficult issues in a SLAM implementation is the data association problem, where the hypothesis of assigning an observation to an existing map location is tested. In our implementation the KLT tracker solves this problem given the high frame rates used to capture images. Since the consecutive images are only 'slightly' displaced, the tracker successfully tracked over 95% of the features in our trial run. However initial SLAM implementation with KLT based data association showed that the EKF based SLAM was not consistent (this is illustrated in Section VI). Further investigations revealed that this was in part due to stereo mismatches and in part due to KLT tracking features inconsistently. Both issues could be attributed to the poor texture presence in the test environment and especially the former issue to the use of a smaller baseline, wide angle camera.

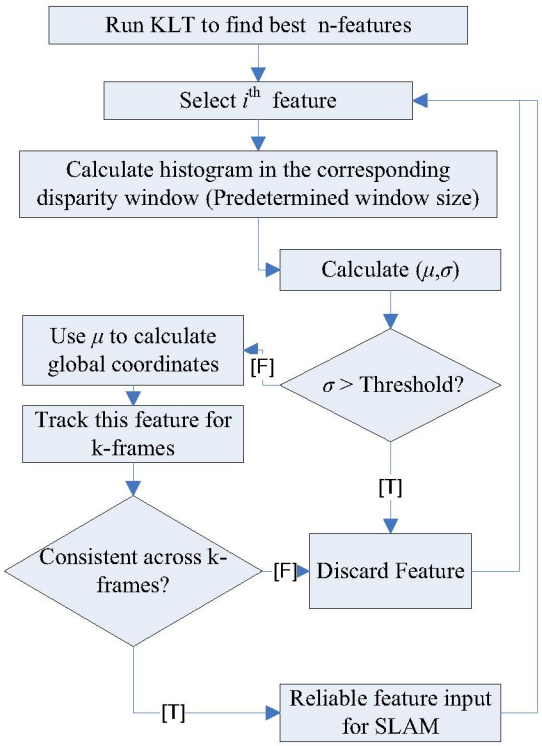

Fig. 2 KLT in combination with the noise filter during feature initialization

In Fig. 3, the left image is a portion of an image taken from a trial run inside our lab. This is a typical image of our test environment where the pathways are very narrow with little texture throughout the camera's field of view. This generates large number of stereo mismatches. The image to the right is the disparity image generated by the stereo correspondence algorithm with lighter shades indicating objects closer to camera. Though there are several small light patches to the middle and right of this image indicating features closer to the camera, in reality these features are quite far from it. This incorrect estimation of disparity in stereo vision gives rise not only to mean shift (gross errors) but also to incorrect estimations of uncertainties (5) in observations. Here we propose a solution to this by utilizing the correlations between features and the camera pose corresponding to a single image. The algorithm begins with the intuition,

Given a single image frame, the features selected are correlated with the camera pose that it was taken. Thus a single successful update in the pose estimate using any of these features should reflect this fact resulting in very small innovations for the remainder of features observed with previously initialized states.

I.e. given a set of features derived from a single image; a feature is selected randomly to update the prior state estimate at current time step: the primary update. A new prior estimate is derived for the features. Then a very tight gate ($0.8\,\sigma$) is used to validate the remainder of features. If a t (80%) number of features satisfy this test, the corresponding feature set is used to update the state. This procedure is iterated until a solution is found or, if not the set that encompasses the maximum no of conforming features is used for the update, once all the features are exhausted in primary updates. The method has its inspirations in the RanSaC [14] algorithm, hence the term t, the threshold parameter. RanSaC was used in [15] for global localisation in matching groups of descriptors to a global map.

V. Closing the Loop

Loop closing is important as it can significantly reduce the uncertainty of the system. A novel adaptive window positioning method is used in testing the hypothesis of seeing an 'old' feature again at an eminent loop closure. Multiple hypothesis of association are generated for a newly observed feature with a subset of features previously seen and bounded

Fig. 3 Part of an image from the stereo camera (left) and the corresponding disparity image (right). Note the gross disparity errors in the right hand side.

by a distance measure based on the state estimates of 'old' features and the estimated position of the new feature. Then each hypothesis is tested with the KLT based dissimilarity measure by adapting the KLT search window position to accommodate the hypothesised feature location. If a unique solution is found for a given dissimilarity threshold corresponding hypothesis is accepted. Else the feature is initialised. Essentially this method relaxes the small inter-frame displacement assumption in the original KLT formulation.

Fig. 4 illustrates this process. Image on the top left shows a new feature found by the KLT feature selection criteria (red dot on the black square on the far wall-magnified in the inset). Right image corresponds to the same part of the environment but taken from a different location during the robot test run. Red dots indicate features selected by KLT. There are three features on the same black square (magnified in the inset) on the wall and this subset of features form three hypotheses according to the above criteria. Bottom image shows that ones the dissimilarity criterion in KLT is applied to them it picks up the feature correctly. This shows that the loop closing is possible with a reasonable viewpoint variance in the two images.

However a prevailing shortcoming in the current method of using KLT based loop closure detection is its inability to handle larger view point variations. This is mainly attributed to the KLT's poor affine and scale invariance properties.

VI. Experimental Results

A. *Experimental Setup*

A Pioneer robot equipped with a MEGA-DCS stereo camera (see Fig.12) from Videre Design is driven through an arbitrary path in an office like environment while the camera is capturing stereo images at 4Hz. Pioneer is also equipped with a SICK laser, which is used separately to capture range and bearing to a set of laser beacons laid in the environment. These observations are utilized in a separate 2D SLAM algorithm for comparison purposes. In this experiment, the number of features selected in each image was limited to 20. This number was kept constant by replacing lost features when tracking of features between images failed. Three different SLAM algorithms were implemented in this experiment and we will reference them later by the names given below.

1. SLAM3D_batch - 3D feature based EKF SLAM with innovation gating to validate KLT tracked features with a batch update [11] algorithm
2. SLAM3D - 3D feature based EKF SLAM with the previously described feature validation algorithm. Utilizes a batch update when the final successful feature set is found.
3. A laser based 2D SLAM algorithm was used for benchmarking the results. It is worth noting that although we consider the laser based SLAM provides the true path, it has a gross error of about 5cm in both cross-track and along-track directions.

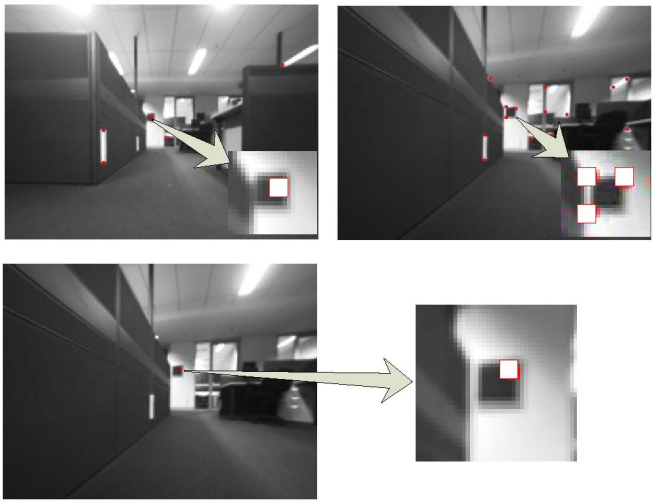

Fig. 4 Multiple hypothesis approach and adaptive window positioning. A new feature detected in the current image (left-inset) and the hypotheses generated (right-inset) and a unique solution is found (bottom).

Same tuning parameters were used for both vision based SLAM algorithms.

B. *Experimental Results*

Fig. 5 shows the robot pose error plots using the SLAM3D_batch along with the 2-sigma error bounds estimated by the EKF. Previously mentioned laser based SLAM was used to generate the 'true' path. It is apparent (by the under estimation of error) that even with good data association provided by KLT the filter is inconsistent.

Fig. 6 shows the results of the same dataset as above with the second algorithm (SLAM3D) without loop closure. Both noise filters discussed in section IV are implemented in this algorithm. The improvement in filter performance is readily noticeable indicating a well tuned EKF. The three-dimensional map generated by the algorithm is shown in Fig. 7. The 2D path estimated from the laser based SLAM is shown in black and light-green line indicates the SLAM3D estimate of the path. It can be seen qualitatively that the estimated robot path from the SLAM3D agrees closely with the more standard 2D laser based SLAM. The errors are only apparent in the final leg of the robot path. Those errors are

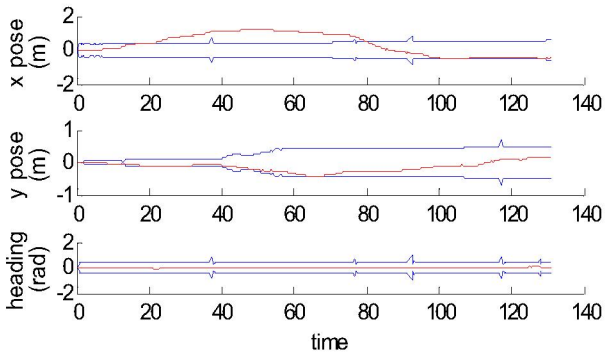

Fig. 5 Robot pose estimate error for SLAM3D_batch relative to the laser based SLAM estimate with the 2-sigma error bounds

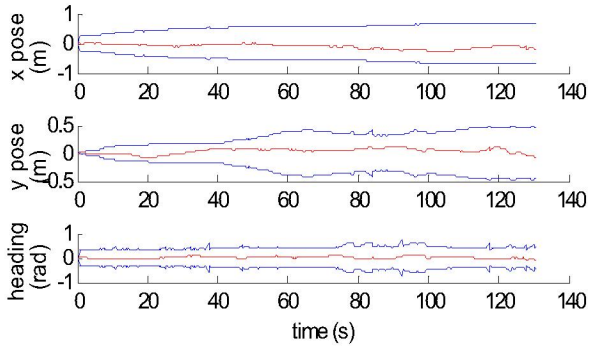

Fig. 6 Relative error between SLAM3D estimated robot pose and the laser based SLAM pose estimate with the 2-sigma error bounds. (without loop closure)

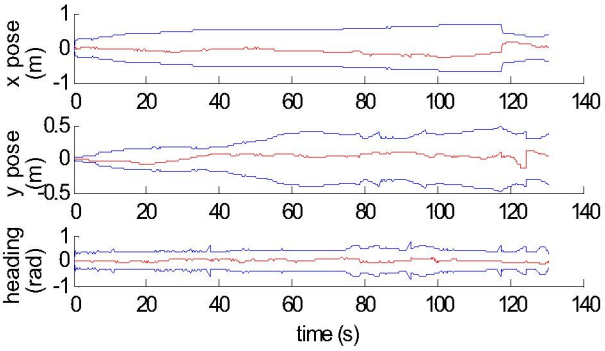

Fig. 8 Relative error between SLAM3D and the laser based SLAM pose estimates along with the 2-sigma error bounds. (with loop closure)

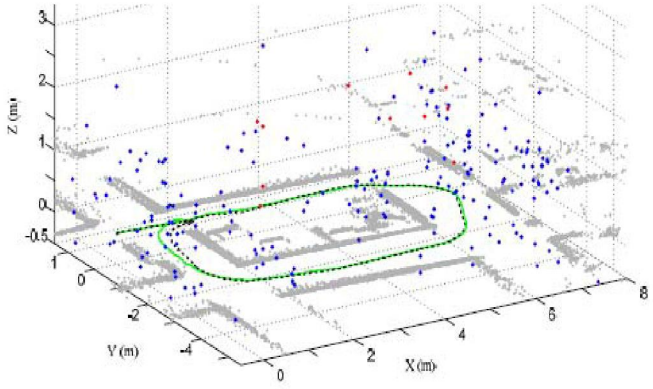

Fig. 7 Map and the estimated path (light-green) along with the laser based EKF SLAM estimate of the path (dotted). (Without loop closure)

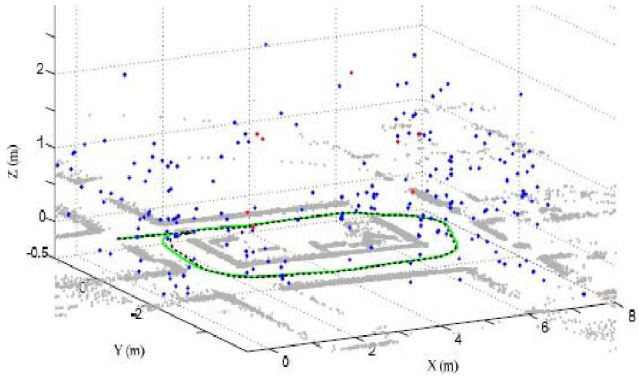

Fig. 9 Map and the estimated path (light-green) along with the laser based EKF SLAM estimate of the path (dotted). (with loop closure)

mainly due to the accumulation of SLAM errors in the long run and lower number of good features registered in some portions of the run (discussed later in the section). As apparent from the misalignment of the wall along the y-direction near x=2 cumulative gross error is prominent in the x-direction while errors in orientation and y-direction are considerably lower.

Fig. 8 plots the state errors and 2-sigma error bounds using SLAM3D with the previously discussed method of loop closure. The loop closure is indicated by the large uncertainty reduction especially in the x and y-directions. Since even without the loop closure the gross errors in the heading estimate were smaller (Fig. 6) it is expected to see only a minor improvement in the relative error in estimates between SLAM3D and the laser based benchmarking algorithm. Fig. 9 is indicative of a successful loop closure with the apparent lack of misalignment of the wall that was present in Fig. 7.

Fig. 10 indicates the feature survival histogram. It could be noted that some good features could last more than 60 frames, which is desirable for SLAM, whilst the others can disappear instantly especially during the third leg of the traverse only a few features have been registered. This is mainly due to KLT picking up features far away along the corridor (due to lack of well textured features nearby) which the stereo algorithm was not able to register proper disparities hence the noise filter has attenuated them. In such ill-conditioned situations odometry contributes locally.

C. *Real-time Implementation*

Currently we have an implementation of the discussed algorithms in real-time sans the loop closure on a pioneer (Fig. 11 & 12). The interface provides a real-time Occupancy grid (Fig. 11) based on the pose estimates from the SLAM algorithm and the laser observations. This provides a visual feed back of the performance of the SLAM algorithm.

We have conducted several test runs in our lab area with statistically reliable results. Table I shows the prominent parameters involved in this implementation. However

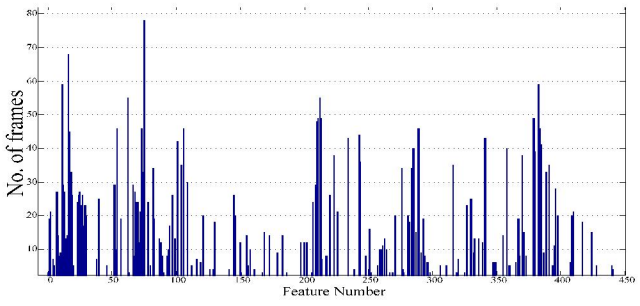

Fig. 10 Feature survival histogram

TABLE I
REAL-TIME IMPLEMENTATION PARAMETERS

Parameter	Value
Camera baseline/(m)	0.088
Image resolution/(pix^2)	320x240
Robot speed/(m/sec)	0.2~0.3
$\sigma_d,\sigma_u,\sigma_v$/(pix)	1.5,1.0,1.0
Processing power	Intel Pentium M, 1.13GHz, 512 MB RAM

depending on the quality of the features picked up by KLT during some of the trials we have observed large absolute errors (>3%). This is suggestive of the necessity for more experiments and further improvements to the algorithms.

VII. CONCLUSION

Stereo vision based SLAM is still a challenging problem due to the gross errors in depth map generation, feature occlusions and other factors present in the environment. We have presented a comprehensive set of methods that has roots in both image processing and Bayesian estimation domains to achieve consistent small baseline stereo vision based SLAM. We believe that to achieve robust SLAM solutions using vision sensors requires an augmentative approach between these two domains. We have presented a methodology for determining good features with the aid of KLT. Further, the data association problem was minimized by utilizing the image feature tracking capability of the KLT. A RanSaC inspired algorithm in the data association phase was also presented. The loop closure was addressed by utilizing a multiple hypothesis filter based on the KLT dissimilarity criterion. This is fairly robust to scale and affine changes. Experiments were carried out in an office like environment to assess the robustness and they show that the filter is consistent. Finally we have presented the real-time implementation of the discussed algorithm without loop closure.

We are furthering our investigations to improve the loop closure with larger view point variations which the current version is not capable of. Especially in ways of utilizing the knowledge of estimated state of the robot and map. We are also in the process of integrating a real-time loop closure method in to the interface.

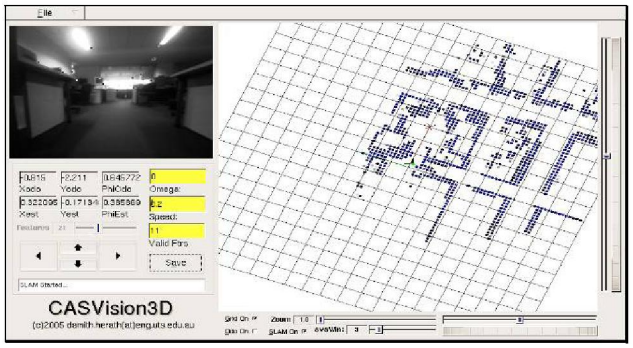

Fig. 11 Stereo vision based real-time SLAM interface

Fig. 12 Pioneer equipped with various sensors including the stereo camera

ACKNOWLEDGMENT

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

REFERENCES

[1] M. Kolesnik, "Vision and Navigation of Marsokhod Rover," presented at ACCV, 1995.

[2] S. Se and M. Brady, "Stereo Vision-Based Obstacle Detection for Partially Sighted People," *Third Asian Conference on Computer Vision (ACCV '98)*, pp. 152-159, 1998.

[3] A. J. Davison and N. Kita, "3D Simultaneous Localisation and Map-Building Using Active Vision for a Robot Moving on Undulating Terrain," presented at IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), 2001.

[4] S. Se, D. Lowe, and J. Little, "Mobile Robot Localization And Mapping with Uncertainty using Scale-Invariant Visual Landmarks," *International Journal of Robotic Research*, vol. 21, 2002.

[5] Davison, A.J., Murray, and D.W., "Simultaneous localization and map-building using active vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 865 - 880 2002.

[6] Murray, D., Jennings, and C., "Stereo vision based mapping and navigation for mobile robots," presented at IEEE International Conference on Robotics and Automation, 1997.

[7] N. M. Kwok and G. Dissanayake, "Bearing-only SLAM in Indoor Environments Using a Modified Particle Filter," presented at Australasian Conference on Robotics & Automation, Brisbane, 2003.

[8] I. K. Jung, "Simultaneous localization and mapping in 3D environments with stereovision," in *LAAS*, vol. PhD. Toulouse: Institut National Polytechnique, 2004, pp. 118.

[9] J. Shi and C. Tomasi, "Good Features toTrack," presented at IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '94) Seattle, 1994.

[10] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem," *IEEE Transactions On Robotics And Automation*, vol. 17, pp. 229-241, 2001.

[11] C. Tomasi and T. Kanade, "Detection and tracking of point features," Carnegie Mellon University CMU-CS-91-132, April 1991 1991.

[12] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," presented at International Joint Conference on Artificial Intelligence (IJCAI '81), Vancouver, BC, Canada,, 1981.

[13] S. Birchfield, "Derivation of Kanade-Lucas-Tomasi Tracking Equation," birchfield97derviation.pdf, Ed., 1997.

[14] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, pp. 381 - 395, 1981.

[15] S. Se, D. G. Lowe, and J. J. Little, "Vision-based global localization and mapping for mobile robots," *IEEE Transactions on Robotics*, vol. 21, pp. 364 - 375, 2005.