3-D Recognition of Remote Objects Utilizing Stereo Vision on a Roving Platform

Mike Wingate
School of Communications and Informatics
Victoria University
Melbourne, Australia.
Michael.Wingate@vu.edu.au

Andrew J Davison
Robotics Research Group
Dept. of Engineering Science
University of Oxford, UK.
ajd@robots.ox.ac.uk

Abstract

An application and supplementation of readily available software to the 3-D recognition and pose determination of scene objects, and navigation of a mobile-robot transporting a stereo head, is described. Initially individual object-models are created by the integration of a sequence of 3-D models constructed from stereo image pairs. An object of interest to be located in a scene is retrieved from a database and matched against scene models until recognition and pose is established. Scene-models are formed while panning the scene. Recognition is based on verifying the existence of mutual groups of 3-D line and or conic edge features in both the scene and model object. Where the object has sufficient distinctive features, recognition is view independent and tolerant to both scale variations and occlusions. Robot navigation is based on both odometry and mapped navigation-features. During navigation, automatic navigation-feature selection and measurement is performed and the results used to substantiate or correct odometry readings.

Key Words: 3-D modeling, recognition, stereo vision, mobile robot navigation.

1. Introduction

A considerable collection of literature exists on 3-D recognition and mobile robot navigation, but few publications have been written on the integration of these two disciplines to form practical working systems. This paper describes the application, supplementation, and integration of two readily available software sources (TINA[1] and SCENE[2]) to produce a mobile robotic system capable of searching for and recognizing 3-D objects in a domestic scene. The mobile robot transports a 4 DOF stereo head equipped with 2 calibrated CCD video cameras and is also able to determine the pose of the object following its location and recognition.

This work is part of an ongoing project concerned with the retrieval of objects in both domestic and industrial environments by an autonomously roving mobile robot. Potential applications range from simple domestic and industrial "robotic aids", to "assistants" for the severely physically disabled or visually impaired.

The paper is divided into 2 main parts: "object recognition", and "robot navigation", both of which constitute an integral part of an operative system. Section 2 discusses implementation of object and scene model building, including a "Model Editor", Section3 recognition (formulation and methodology), and Section 4 navigation and localisation of the robot (algorithms and methodology). A number of results are displayed in Section 5.

Initially individual object-models are created by the integration of a sequence of 3-D models constructed from stereo image pairs, each representative of the object in varying angular positions. A computer driven horizontal circular table is provided for this process. Robot navigation is based on both odometry information and mapped visual-features. Scanning of the scene for objects to be recognised occurs at specified locations termed waypoint stations.

2. Model Creation

If two images of a scene are captured using a pair of calibrated cameras, edge based binocular stereo triangulation can be used to obtain a partial 3-D edge outline of the scene as depicted in the left (or right) camera image. This is of limited use for 3-D recognition as the object (or scene) is only depicted from one view. For example if a solid cube object has letters 'A' printed on its front and 'B' on the backside, the letter 'B' may not be visible from a particular view. If both letters were required for recognition the block would not be identifiable. A 3-D outline of the full cube (i.e. hidden edges included) on the other hand, would ensure all its surface features were visible and would thus constitute a suitable object-model for recognition.

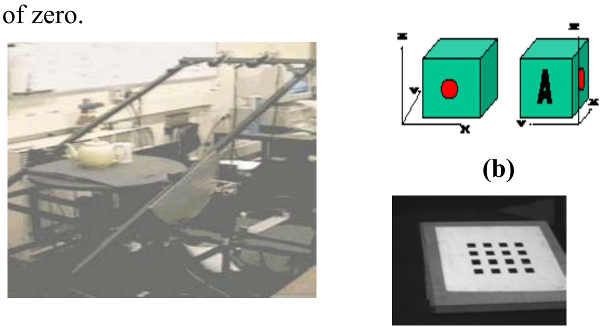
To facilitate 3-D line model creation, a computer driven rotating table with two Hitachi video Cameras mounted on an extended frame (Fig 2.0.1 a) was used to capture stereo pairs of multiple views of desired objects.

[1] TINA is a machine vision research environment running under Unix (X-windows). It is written in C and includes a substantial collection of integrated image processing packages.
Source: http://www.niac.man.ac.uk/Tina/

[2]SCENE is a flexable open-source C++ library for sequential localisation and map-building. It is highly suited for 2D/3D robot navigation. Source: http://www.robots.ox.ac.uk/~ajd/Scene

From these a 3-D partial edge outline of each view was derived and ultimately integrated to form a complete model.

The coordinate frame for each view was (by default) with respect to the left camera's optical centre. Merging the independent views requires all re-constructions to be with respect to a unique (although arbitrary) co-ordinate frame attached to the object. As the object rotates with the table, so too does its coordinate frame, thereby ensuring all the object's physical dimensions and features remain static with respect to that frame (Fig 2.0.1 b). These would however change with respect to any stationary reference frame. The World reference was chosen so that its Z-axis coincided with the table's center axis of rotation. The object's arbitrary co-ordinate frame was assumed aligned with the World coordinate frame for a table rotation angle of zero.



(a) **(b)** **(c)**

Fig. 2.0.1 (a) Models created on rotating table
(b) Features remain static w.r.t objects's frame
(c) Camera calibration tile

As the final object's reconstruction is required with respect to its own coordinate frame, two transformations of 3-D edgels are required. The 1st from the left camera coordinate frame to the static world reference frame, and then a simple rotational transformation (about the world Z-frame.

Calibrating the cameras using a calibration tile (Fig 2.0.1 c) whose Z-axis also coincided with the table's center, and a knowledge of subsequent angles of table rotation, enabled the required 4x4 (homogeneous) left camera to world, and world to object transformation matrices to be determined. Each edgel of the reconstructions was subsequently transformed to the desired rotating reference frame and combined to form the 3-D models. Fig. 2.0.2 depicts a line frame model of a cup and an alphabet cube respectively.

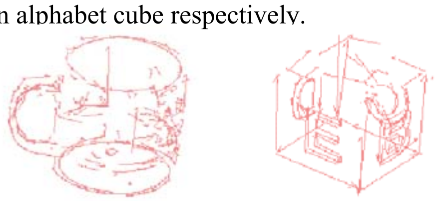


Fig. 2.0.2 Line frame model of cup and cube

The above procedure was coded and added to the TINA's smm_tool source code module [1].

Initially, three objects, an alphabet cube, a mug, and a teapot were produced for test purposes. These models comprised a database of objects to be recognized in a scene. Fig. 2.0.3 shows the stereo images of the scene and subsequent scene-model produced from the two images.

(a) *(b)*

Fig 2.0.3 (a) Stereo images of scene (b) 3-D scene-model

2.1. Model Enhancement – Model Editor

A Model Editor based on TINA's geomstat tool [3] was developed to facilitate the manual removal and/or addition of 3-D segments in formed models. Editing of models is desirable on occasion as double lines or conics arise or are fragmented or omitted altogether due to camera lens distortions, calibration inaccuracies, occlusions, etc.

An example of the editor's application to one of our test cubes is given in Fig 2.1.1

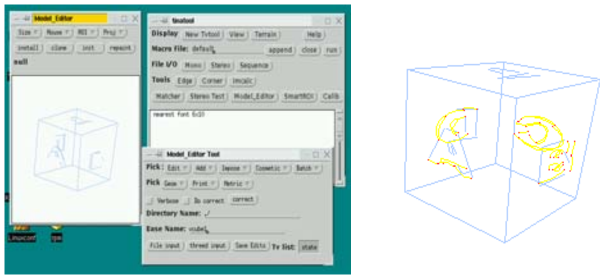


Fig 2.1.1 Application of model editor

3. Recognition Algorithm

The recognition algorithm, incorporated in TINA 's Matcher_tool module and written by Pollard et al [1]. [2] was supplemented by the addition of code to automate the recognition process. This was necessary to facilitate autonomous operation of the robot. Recognition is based on pairwise matching of primitive geometrical features associated with a model and the scene. These features comprise 3-D line segments and or 3-D conic segments.

Straight-line segments are represented by the quadruple ($\mathbf{l}_1$, $\mathbf{l}_2$, $\mathbf{e}_1$, $\mathbf{m}_1$). That is their two end points $\mathbf{l}_1$ and $\mathbf{l}_2$, the direction vector between them $\mathbf{e}_1$ and centroid of the line, its midpoint $\mathbf{m}_1 = (\mathbf{l}_1 + \mathbf{l}_2)/2$.

In the case of line segments, three pairwise relationships are used. These are (a) orientation, (b) minimum distance apart, and (c) distance from minimum distance apart. These are described more fully below and illustrated in Fig. 3.0.1.

Orientation difference between two pairs of potential matches: This is obtained from the dot product of $\mathbf{e}_1$ and $\mathbf{e}_2$ ie $\alpha = \cos^{-1}(\mathbf{e}_1 \cdot \mathbf{e}_2)$.

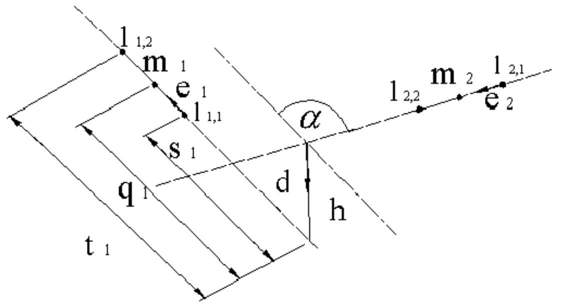


Fig 3.0.1 Matching of line segments

Minimum distance apart between (extended) lines:
This is represented by the unit vector (normal to each line) d, where:

$$\mathbf{d} = (\mathbf{e}_1 \mathbf{x}\ \mathbf{e}_2)/|\mathbf{e}_1 \mathbf{x}\ \mathbf{e}_2|$$

and the scalar distance between the lines h, where:

$$h = (\mathbf{m}_2 - \mathbf{m}_1) \cdot \mathbf{d}.$$

If the line segments are close to parallel the distance used is the perpendicular distance between the lines:

$$h = |(\mathbf{m}_2 - \mathbf{m}_1) - [(\mathbf{m}_2 - \mathbf{m}_1) \cdot \mathbf{e}_1]\mathbf{e}_1|,$$

while for non-parallel lines, the distances along the lines to the start and finish of each line from the point of minimum distance apart is used. From the above the vector between the points of minimum distance apart is given by

$$\mathbf{h} = h\mathbf{d}.$$

Moving $\mathbf{m}_2$ to $\mathbf{m'}_2$ by adding $-\mathbf{h}$ produces a line ($\mathbf{e}_2$, $\mathbf{m'}_2$) such that lines ($\mathbf{e}_1$, $\mathbf{m}_1$) and ($\mathbf{e}_2$, $\mathbf{m'}_2$) are coplanar and meet at the point of minimum distance apart on ($\mathbf{e}_1$, $\mathbf{m}_1$).

$$q_1 = \{(\mathbf{e}_2 \mathbf{x}\ \mathbf{e}_1) \cdot [\mathbf{e}_2 \mathbf{x}\ (\mathbf{m'}_2 - \mathbf{m}_1)]\} / |\mathbf{e}_2 \mathbf{x}\ \mathbf{e}_1|^2$$

is the signed distance to that point from m_1 in the direction e_1. Distances from $\mathbf{l}_{1,1}$ and $\mathbf{l}_{1,2}$ to that point are given by

$$s_1 = q_1 + (\mathbf{m}_1 - \mathbf{l}_{1,1}) \cdot \mathbf{e}_1 \text{ and}$$
$$t1 = q_1 + (\mathbf{m}_1 - \mathbf{l}_{1,2}) \cdot \mathbf{e}_1, \text{ respectively.}$$

Similarly the distances to the point of minimum distance apart along line2 are

$$s_2 = q_2 + (\mathbf{m}_2 - \mathbf{l}_{2,1}) \cdot \mathbf{e}_2 \text{ and}$$
$$t_2 = q_2 + (\mathbf{m}_2 - \mathbf{l}_{2,2}) \cdot \mathbf{e}_2$$

Prospective matches for each pair of elements

$$\{(s_i,t_i), (s_{i+1},t_{i+1})\}$$

from the object model description can be tested for geometrical correspondence with each pair of elements in the scene model.

3.1. Stored Table of $\{(s_i,t_i), (s_{i+1},t_{i+1})\}$

An advantage of the use of the pairwise elements $\{(s_i,t_i), (s_{i+1},t_{i+1})\}$ is that they can be precalculated separately for each pair of lines and stored in look-up tables. To accommodate errors and scale differences, a range for overlap is provided. These are accounted for as follows.

Pairs of lines are allowed errors

$$|\eta_1| < \beta_1 \text{ and } |\eta_2| < \beta_2$$

on the location of their centroids ($\mathbf{m}_1$ and $\mathbf{m}_2$) and direction vectors ($\mathbf{e}_1$, $\mathbf{e}_2$). The latter allowable errors are in terms of solid angles (θ_1 and θ_2). On orientation differences, the interval is

$$[\max(\alpha - \theta_1 - \theta_2, 0), \min(\alpha + \theta_1 + \theta_2, \Pi)],$$

while for the minimum distance apart between (extended) lines, the interval is

$$h \pm (\beta_1 + \beta_2 + |q_1| \tan \varphi_1)$$

For s_1, t_1, and s_2, t_2 the permissible range is:

$$s_1 \pm (\beta_1 + \beta_2 + |q_2| (\tan \theta_2/\sin \alpha))$$
$$t_1 \pm (\beta_1 + \beta_2 + |q_2| (\tan \theta_2/\sin \alpha))$$
$$s_2 \pm (\beta_1 + \beta_2 + |q_1| (\tan \theta_1/\sin \alpha))$$
$$t_2 \pm (\beta_1 + \beta_2 + |q_1| (\tan \theta_1/\sin \alpha))$$

A similar range of overlapping intervals is provided for pairs of conic features (i.e. arc lengths and radii)

Initially, partial matches of feature elements between the object and scene are formed into groups called cliques. Each clique has a "focus feature"

3.2. Focus Features

Focus features provide a basis for determining maximal consistency of neighbouring elements within a clique. For object models the cliques are manually chosen and will generally consist of a distinctive focus feature (such as an arc length or long line) and a number of surrounding local features in close proximity. Object models in general will comprise sufficient cliques so that adequate features can be matched in a scene irrespective of the objects orientation.

Matching is performed as follows:

Object model focus features are selected in sequence. Closest matches to the selected focus feature in the scene model are considered as potential matches. Neighbouring features are considered in terms of their pairwise geometrical relationships.

Cliques are ranked according to the highest number of neigbourhood matches consistent with the object model group size. Transformations to place each found clique into its corresponding position in the scene is calculated using the method described by Faugeras et al [4]. Cliques that produce near identical transforms are considered as belonging to the same object.

The mean transform is returned to quantify the pose of the object in the scene, however only the 3x3 rotation matrix is pertinent as translation is referenced to the frame used during model creation. A translation with respect to the left camera can be determined from the xyz segment values of the scene model.

The sequence for matching a model with a scene is:

(i) Build the pairwise geometric tables.
(ii) Perform matching of cliques
(iii) Compute the transform of the object model to carry it into the scene.
(iv) Transform the model into the scene (placing and displaying it over the recognized object).

Associated with the matching process are seven parameters whose values can be adjusted to reflect the confidence in the scene data namely:

(a) Clique size, (b) Position error, (c) Rotation , (d) Length threshold, (e) Maximum rotation, (f) Maximum translation, (g) Length ratio.

3.3. Automation of the Recognition Process

To automate the recognition process the "current matching sequence" mentioned above was formulated into a simple iterative loop in which the seven parameters are systematically modified to reflect a reduction in data "confidence levels". Iteration continues until either recognition occurs, at which stage the loop is exited, or the set number of iteration step levels (currently 12) is exceeded and recognition is deemed as failed. If recognition failure occurs the next captured scene model is entered into the loop. This process proceeds as the pan-tilt-vergence head sweeps through a pan arc of –20 to +20 degrees), tilt and vergence angles being held fixed. This is repeated at each scheduled observation station (waypoint).

4. Mobile Robot Navigation

The navigation software is based upon "SCENE: Generalised Software for Sequential Map-Building and Localisation" [7][8]. The map-building implemented in SCENE uses an Extended Kalman Filter with full covariance coupling between robot and navigation-feature states.

For flexibility "SCENE" allows the user to develop and incorporate a motion model tailored to a specific robot-base. While several models were considered, a two stage model consisting of a pure rotation (about the base's central axis) followed by a translation was chosen (fig 4.0.1).

In addition, interface and communications software was added to permit dialog with the frame grabber, pan-tilt-vergence head, and the robot-base's controller. This proved to be relatively simple utilizing "interface text" files, the contents of which signal appropriate requests for action or data.

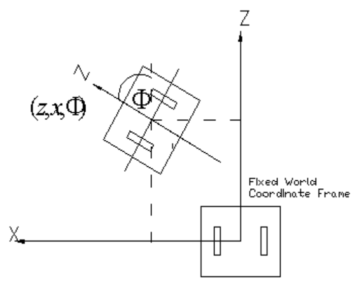


Fig 4.0.1 Coordinate Frames : the robot's 2-D position in the world is represented by the vector (z,x,Φ)

In this application autonomous path-planning is not required as navigation proceeds via a sequence of pre-specified observation stations called waypoints. The robot is programmed to travel to selected waypoints, during motion the head gazes upwards and searches for known or new navigation-features. Known features are read into SCENE on initialisation of the software and comprise of image patches of the features as well as their xyz positional information.

The process of acquiring and storing known features prior to navigation can be considered part of a map-building process and consists of driving the robot to points on route (as well as waypoints), and activating SCENE's "Initialise Automatically-Selected Point Features" function. If a suitable feature is found, the head is driven by the function until "Lock On[3] " is achieved, and the x,y,z of the centre of the feature returned (in vehicle co-ordinates). The image of the left patch is subsequently archived as a "known_patch" and the xyz positional information stored as a "known_feature". Examples of features could be ceiling rose segments, fluorescent light housing parts, wall marks, elevated signs etc, (see fig 5.2.1).

At regular steps during navigation, head fixation angles for the nearest known_feature are calculated (from stored xyz positional information), and the head driven to gaze in the direction of the known_feature. A search for a corresponding "patch match" (within a narrow spatial window whose size is determined by map uncertainty) ensues.

Matching is carried out by a correlation search: the region of the search window correlating best with the saved feature-patch is found. This automatic navigation-feature selection and measurement is performed at each step interval and the results of measurements used to substantiate or correct odometry readings via the EKF. This ensures that the robot has confidence in the knowledge of its whereabouts, even if wheel slippage should occur.

Following localisation, at waypoints associated with object locations (i.e. tables laden with objects), the head's gaze is lowered and it sweeps through a pan of –20 to +20 degrees in search of the desired object. At present the tilt and vergence angles remain fixed during this sweep. This ensures the cameras' motion (and consequently the structure-from-motion) is consistent with their calibration. It is proposed to vary the tilt and vergence angles in a sequence of movements in the future for a variety of different calibrated camera motions.

4.1. SCENE's Map-Building and Localisation Algorithm

4.1.1. Visual Landmarks

The basis for localisation using vision is a map of features which are repeatably measurable using the robot's cameras. Within a general mapping framework, there is the potential for these features to have many different forms: points, lines or planes for instance. In our current

[3] "Lock On" occurs when the image of the centre of the selected patch is at the principal point of both the left and right cameras

implementation, point features in 3D space are recognised using image correlation matching, which proves to be surprisingly robust to changes in viewpoint. The active head moves to fixate features for measurement with both of its cameras, acting as an accurate "pointing stick" which can measure the direction and depth of features over a very wide field of view.

4.1.2. Storing and Updating Map Data

The current state of the robot and the scene features which are known about are stored in the system state vector $\hat{\mathbf{x}}$ and covariance matrix P. These are partitioned as follows:

$$\hat{\mathbf{x}} = \begin{pmatrix} \hat{\mathbf{x}}_v \\ \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 \\ \cdot \\ \cdot \\ \cdot \end{pmatrix}, \qquad P = \begin{bmatrix} P_{xx} & P_{xy_1} & P_{xy_2} & \cdots \\ P_{y_1x} & P_{y_1y_1} & P_{y_1y_2} & \cdots \\ P_{y_2x} & P_{y_2y_1} & P_{y_2y_2} & \cdots \\ \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \end{bmatrix}$$

$\hat{\mathbf{x}}_v$ is the robot state estimate, and $\hat{\mathbf{y}}_i$ the estimated state of the ith feature ($\hat{x}, \hat{y}, \hat{z}$ for the case of a 3-D point). The state vector represents the robot's map of its environment and its place within it, and the covariance matrix how uncertain this Feature positions may be supplied as prior information when the robot starts navigating, or dynamically added to or removed from the map during navigation as required.

A full SLAM[4] approach is used, propagating the covariances between the robot state and all feature estimates, and between the feature states themselves. This is essential in our system, since the small number of features generally used must have estimates which are of very high quality to provide accurate localisation information.

The ability of active cameras to view features over a huge field of view is key to the quality of this map: the robot can really see the same features continuously as it goes through very large motions and rotations; thus fewer features need to be added to the map, and the uncertainties related to those present can be reduced succesively as the robot is able to measure them repeatably over long periods.

The data is updated sequentially as the robot moves around its environment and makes measurements of the features in its map following the rules of the Extended Kalman Filter: a prediction step when the robot moves, when a new position estimate is calculated based on odometry, and an update step when a measurement is made of a feature. Our map-building software system supports plug-in models which describe the specifics of robot motion and feature measurement.

[4] Simultaneous localisation and map-building.

4.2. Active Measurement

In an active scenario, it is necessary to decide at each instant which feature in the map to attempt to measure. This decision is made based on two criteria: expected visibility and the value of the measurement. The expected visibility (more precisely measurability) is something that depends on the sensor and feature type: for instance, with our point features matched by correlation, we do not expect to be successful with matching if the viewpoint is too different from that from which they were initially seen. Since we have an estimate of the current robot position, the predicted viewing direction can be evaluated in this respect.

Once a measurable subset of features in the map has been identified, the value of measuring each one is evaluated in terms of the uncertainty of their position relative to the robot (we choose a measurement which has a high innovation covariance, the general principle being that there is little use in making a measurement of which we are sure of the result.

The selected feature is then measured by driving the active head to the angles predicted for fixation on that feature, and searching the images obtained for a match. Precise search regions are calculated from the uncertainty in the map, which maximise computational efficiency and reduce the chance of mismatches.

5. Results

5.1. Object Recognition & Location

Fig 5.1.1 depicts a recognized cup and alphabet cube transformed into the scene of Fig 2.0.3. Note that pose and scale have been correctly identified.

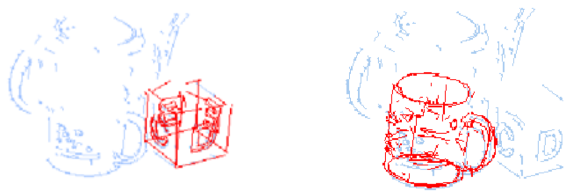


Fig 5.1.1 cup, cube matched, scaled, and placed over object in the scene

In Fig 5.1.2 (a) the robot is at a waypoint panning for the hole-punch. (b) shows the left image of a scan sequence search for the hole-punch, while (c), the retrieved model of the hole punch. (d) shows the scene model produced corresponding to (b), and (e) the model transformed into the scene following recognition and (f) the corresponding transformation matrix (ie pose details).

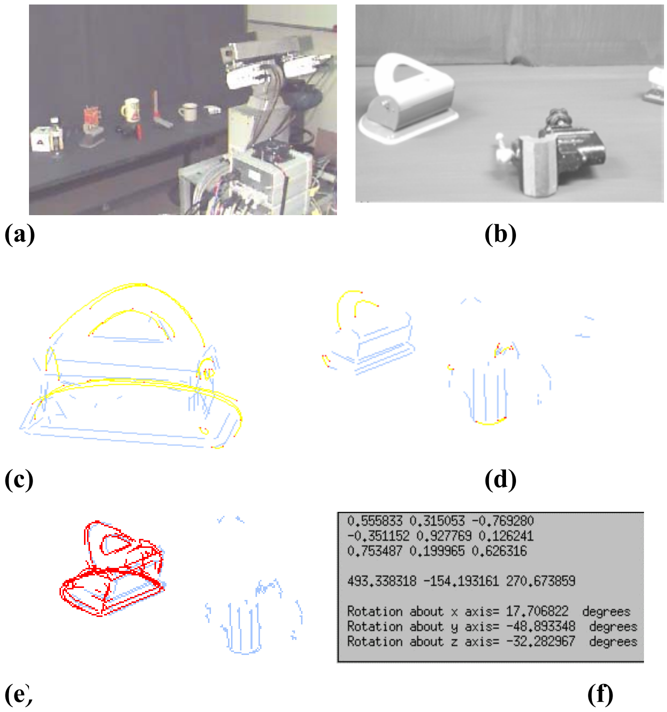
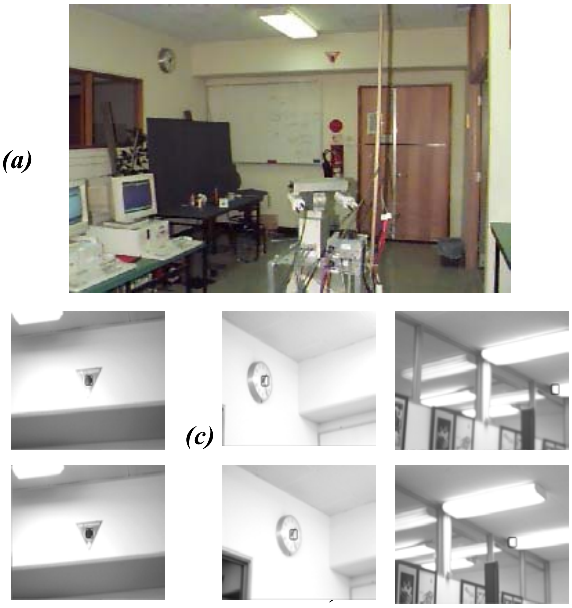


Fig 5.1.2. Hole-Punch Search (a) Panning robot at a waypoint,(b) Table scene: Captured left image, (c) Model of hole-punch(d) Scene-model of (b), (e) Hole-punch found and model transformed into scene, (f) Pose transformation.

5.2. Navigation

The robot navigating its way to the "*table scene*" waypoint, and typical feature-patches are shown in Fig 5.2.1. "Lock onto the *"fire extinguisher sign*" at the waypoint, is depicted in fig (b). This patch (left image), together with coressponding xyz positional data was stored as a known-feature.

(a)



(b) *(c)*

Fig 5.2.1 (a) Robot 1.2 metres from waypoint, (b) known feature-patch. (c) feature patches.

5.3. Processing Times

Image processing at present is slow, essentially because images format differences exist between the two software packages and in addition images are required to be transmitted serially from an on board computer to a stationary one. Both of these overheads will be eliminated in the future. It is worth noting that in [9] from which SCENE was developed, real-time tracking of approx.5 measurements/second was attainable.

6. Acknowledgement

The authors are grateful to P.Rockett, S. Crossley and J. Porrill of the Univ. of Sheffield for their valuable discussions and for providing access to their hardware and software resources.

7. References

[1] Porrill, J, Pollard, S B Pridmore, T P Bowen, J, Mayhew, J E W and Frisby J P, 'TINA: the Sheffield AIVRU vision System'. IJCAI9, Milan, Italy 1987, pp 1138-1144.

[2] S.B. Pollard, T.P.Pridmore, J. Porrill, J.E. Mayhew and J.P Frisby, 'Geomtrical Modeling from Multiple Stereo Views' The Int. Journal of Robotic Research, Vol 8, No4, pp3-19, 1998

[3] S.B. Pollard, J. Porrill and J.E. Mayhew, 'Recovering partial 3D wire frames descriptions from stereo data', Image and Vision Computing, vol 9 No 1 pp 58-65, 1991.

[4] Faugera, O D,Hebert, M, Ponce J, and Pauchon, E 'Object representation, identification, and positioning from range data' Proc 1st Int. Symp. On Robotics Res. MIT Press, Cambridge, MA, USA, 1984, pp 425 – 446.

[5] A.C.Evans, N.A.Thacker and J.E.W.Mayhew, 'A Practical View Based 3D Object Recognition System', IEE conference on neural networks, Brighton, 1993a.

[6] Andrea Selinger and Randal C. Nelson, 'Improving Appearance-Based Object Recognition in Cluttered Backgrounds'', TR725, Computer Science Dept., U. Rochester, January 2000.

[7] A.J.Davison 'Models and State Representation in Scene: Generalised Software for Sequential Map-Building and Localisation' , Feb. 6, 2001: http://www.robots.ox.uk/~ajd/ .

[8] A. J. Davison and N. Kita. 'Sequential localisation and map-building ', Computer vision and robotics. Int. Proceedings of the 2nd Workshop on Structure from Multiple Images of Large Scale Environments (SMILE), in conjunction with ECCV 2000, Dublin, Ireland. Springer-Verlag LNCS, 2000.

[9] A.J.Davison 'Mobile Robot Navigation Using Active Vision', Ph.D. thesis, University of Oxford, 1998.

[10] Schultz, A., Adams, W., and Yamauchi, B., 'Integrating Exploration, Localization, Navigation and Planning With a Common Representation', Autonomous Robots, vol 6 No3, May 1999.IEEE Press, pp. 293-308.