



Building 3-D Visual Perception of a Mobile Robot Employing Extended Kalman Filter

SRIKANTA PATNAIK

Department of Electrical Engineering, University College of Engineering, Burla-768 018, India;
e-mail: srikantapatnaik@hotmail.com

AMIT KONAR and AJIT K. MANDAL

Department of Electronics and Telecommunication Engineering, Jadavpur University,
Calcutta-700032, India

(Received: 26 June 2000; in final form: 31 July 2001)

Abstract. The paper aims at designing a novel scheme for sensory data fusion by a mobile robot for reconstructing its 3-D world from their multiple gray images. Extended Kalman filter has been employed for determining the coordinates of the 3-D vertices and equation of the planes of the obstacles in the robot's workspace from their multiple images. The geometric relations among these 3-D planes are then determined by using a logic program for recognizing the obstacles. The time required for recognition of a typical planer obstacle such as a box on a Pentium-III client with 64 MB RAM and a Pioneer-2 type robot server including the time involvement for the motion of the robot around the obstacle is approximately 18 seconds.

Key words: data fusion, Kalman filtering, 3-D reconstruction, object recognition and logic programming.

1. Introduction

A mobile robot constructs its world map by sensing and processing information about the objects around it for subsequent use in the navigational planning stage [15]. A robot can execute the navigational planning task more efficiently with the foreknowledge about its 2-D and 3-D world. In a 2-D map-building problem, ultrasonic or laser range finders are usually employed to determine the location and boundary of obstacles in the robot's world map. There exist a number of literatures on the two-dimensional Map building [2, 8, 14]. Some of these prefer metric based approach, while the rest employs strategies to identify landmark first and then use local search strategy to explore the unvisited obstacles. Construction of a 3-D world map, however, needs to keep a track of the obstacle surfaces and their heights as well. Two or more cameras mounted on a pan-tilt platform are usually employed to determine the third dimension of obstacles. Such a system with the capability of determining "depth information" from multiple images is called stereo vision [16] system. A low cost mobile robot having a single camera mounted on a pan-tilt platform can also be used for stereo vision if 2 or more successive frames taken at

different camera orientations are used for finding the depth information. The paper aims at developing an experimental scheme for recognizing planer objects around a mobile robot by fusing multi-sensory image data using extended Kalman filtering [10].

A Kalman filter is a recursive digital filter [6] that evaluates estimator(s) from a number of successive measurements supplied to the filter in sequence until the error in the process of estimation, described by a covariance matrix, comes within a prescribed limit. One significant advantage of Kalman filter over least square curve fitting lies in its recursive formulation that supports submission of input data points in time sequence. The Kalman filter can be used to construct 2-D lines from noisy 2-D image points, affine 3-D points from 2-D image points, affine 3-D lines from noisy 2-D image points or from affine 2-D lines or from 3-D points, and 3-D planes either from 3-D points or 3-D lines. Ayache employed Kalman filter [3, 4] for the 3-D reconstruction of images.

In this paper a novel experimental scheme for recognizing planer objects from their multiple images will be presented. The object, if any in a scene, should first be segmented from its background and localized for subsequent image processing. Next, the Laplacian operator will be employed at the localized region on the original image to find the 2-D edges. The 2-D vertices on the common edges will then be traced. The 3-D vertices corresponding to the 2-D vertices will be determined next from multiple images by using Extended Kalman Filtering (EKF). The expressions for 3-D planes then can be determined from multiple 3-D points lying on the planes. Finally, a logic program will be employed to determine relationship among the bounding planes of an obstacle for its recognition.

The paper has been classified into the following 6 sections. A brief outline of the EKF is presented in Section 2. The relationship of the perspective projection matrix with camera parameters such as pan, tilt and skew angles [9] will be discussed in Section 3. The representation of 3-D points and 3-D planes in a minimal form and determination of the filter parameters for the reconstruction of 3-D points and 3-D planes will be presented in Section 4. The experimental details including the setup and results will be presented in Section 5. The conclusions will be listed in Section 6.

2. Recursive Extended Kalman Filter

An extended Kalman filter is a digital filter that attempts to minimize the effect of measurement noise on the estimation process of a given set of parameters, related linearly with the set of measurement variables. The recursive extended Kalman filter is a recursive formulation of the filter algorithm so that the parameters to be estimated called estimators can be evaluated from the measurement inputs supplied to the filter in succession. An estimation error covariance matrix is used to monitor the accuracy in the estimation process at successive iterations of the algorithm. The algorithm is terminated when the elements of the error covariance matrix come

within a prescribed limit. The most important characteristic of this filter lies in its recursive formulation that provides a user the freedom to control the accuracy of estimation to a desired level at the cost of new measurement inputs.

Let

- \mathbf{x}_i be a measurement vector of dimension $(m_i \times 1)$,
- \mathbf{K}_i be the filter gain matrix of dimension $(n \times p_i)$,
- \mathbf{a}_i be the estimator vector of dimension $(n \times 1)$,
- \mathbf{M}_i be a system matrix of dimension $(p_i \times n)$ such that

$$\mathbf{M}_i = \frac{\partial \mathbf{f}_i}{\partial \mathbf{a}}, \quad (1)$$

where $f_i(\mathbf{x}_i, \mathbf{a}) = 0$, be a set of equations describing relationships among a parameter vector \mathbf{a} and the measurement variable vector x_i ,

$$y_i = -f_i(x_i^*, a_{i-1}) + \left[\frac{\partial f_i}{\partial a} \right]_{(x_i^*, a_{i-1})} (a - a_{i-1}^*)$$

is a modified measurement vector of dimension $(p_i \times 1)$, obtained by linearization of $f_i(\mathbf{x}_i, \mathbf{a}) = 0$ around $\mathbf{x}_i = \mathbf{x}_{i-1}$ and $\mathbf{a}_i = \mathbf{a}_i^*$ using Taylor series.

$$\mathbf{w}_i = E[w_i w_i^T] = \left[\frac{\partial f_i}{\partial x} \right]_{(x_i^*, a_{i-1})} \Lambda_i \left[\left[\frac{\partial f_i}{\partial x} \right]_{(x_i^*, a_{i-1})} \right]^T, \quad (2)$$

where $\mathbf{w}_i = (\partial \mathbf{f}_i / \partial \mathbf{x})(\mathbf{x}_i - \mathbf{x}_i^*)$ is the measurement noise vector of dimension $(p_i \times 1)$ and Λ_i is a positive symmetric matrix.

$$\mathbf{S}_i = E[(\mathbf{a}_i - \mathbf{a}_i^*)(\mathbf{a}_i - \mathbf{a}_i^*)^T]$$

is the error covariance matrix of the estimator \mathbf{a} .

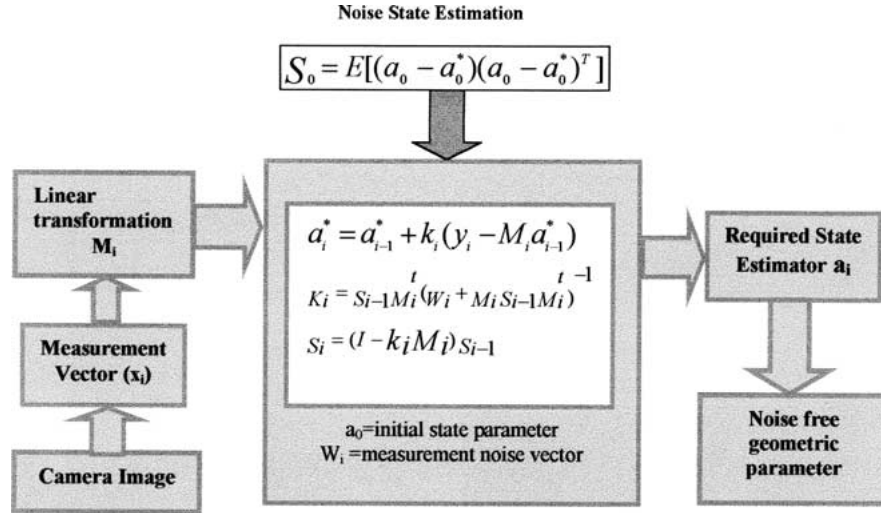


Figure 1. Schematic diagram of a recursive Kalman filter.

The recursive formulation of an EKF includes the following three steps.

$$\mathbf{K}_i = \mathbf{S}_{i-1} \mathbf{M}_i^T (\mathbf{W}_i + \mathbf{M}_i \mathbf{S}_{i-1} \mathbf{M}_i^T)^{-1}, \quad (3)$$

$$\mathbf{a}_i^* = \mathbf{a}_{i-1}^* + \mathbf{K}_i (\mathbf{y}_i - \mathbf{M}_i \mathbf{a}_{i-1}^*), \quad (4)$$

$$\mathbf{S}_i = (\mathbf{I} - \mathbf{K}_i \mathbf{M}_i) \mathbf{S}_{i-1}. \quad (5)$$

The algorithm is initialized with a large \mathbf{S}_0 . The values of \mathbf{y}_i , \mathbf{M}_i , \mathbf{W}_i are computed following their above definitions. \mathbf{a}_0 is initialized as a null vector. The algorithm then continues iterating in sequence until \mathbf{S}_i comes below a predefined threshold. The resulting \mathbf{a}_i after termination of the algorithm is the desired estimator. The schematic diagram depicting the use of EKF in estimating noise-free geometric parameters from noisy 2-D images is presented in Figure 1.

3. The Perspective Projection Geometry

The back plane of a pinhole camera where the image of an object is formed is called the **image plane**. From the principle of perspective projection it can be easily shown that the relationship between a bright spot formed at (u, v) on the image plane corresponding to an object point (x, y, z) (*vide* Figure 2) is given by

$$\begin{pmatrix} ku \\ kv \\ k \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{f} & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \quad (6)$$

In expression (6), k is a scalar constant given by $k = -z/f$, and f is the focal length of the camera. In the above system the coordinates (u, v) and (x, y, z) are referred to with respect to the image plane axes X_1 and Y_1 and the z -axis Z_c perpendicular to the plane. If the origin of that system is shifted to the optical center of the camera, then the transformation can be represented by

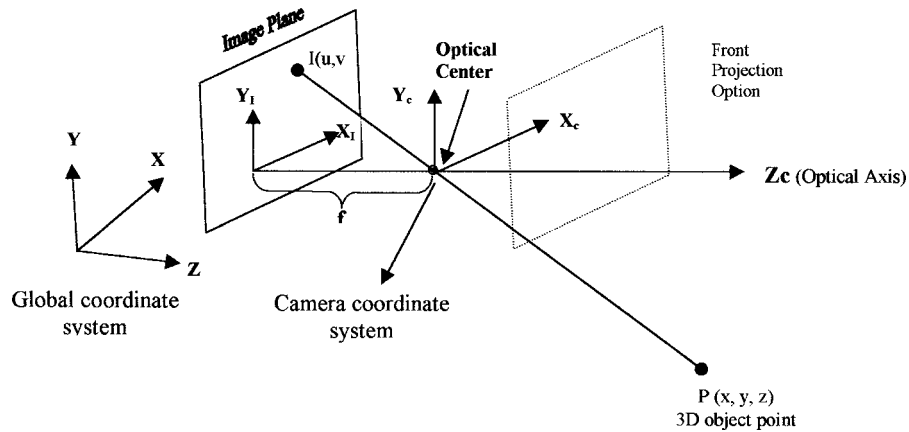


Figure 2. Generalized camera model of 3-D projection.

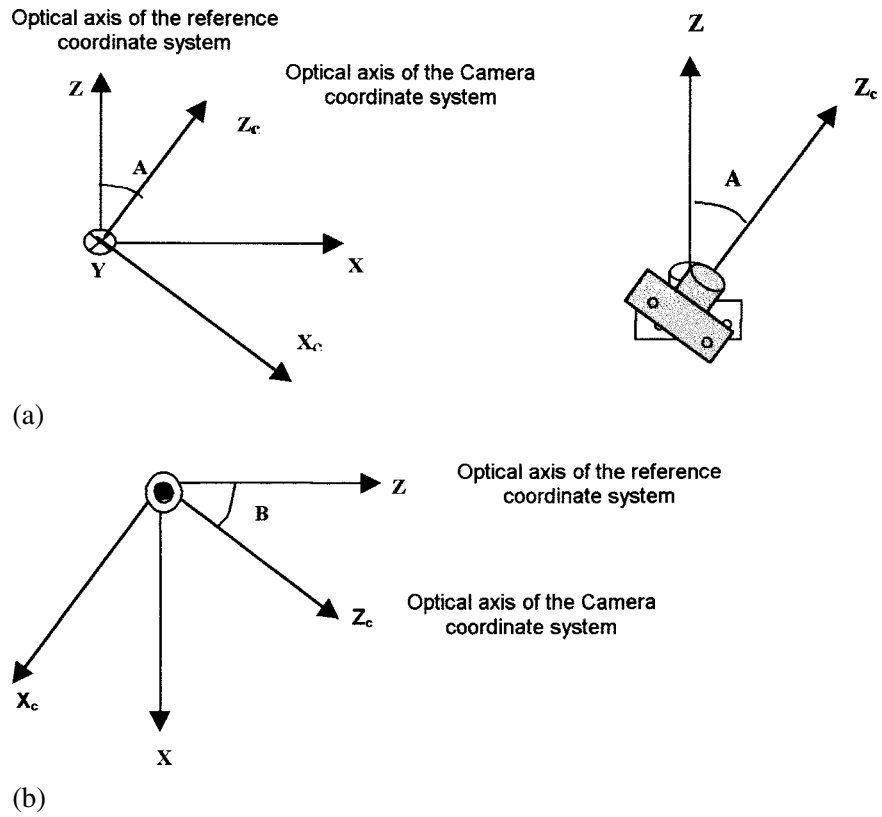


Figure 3. (a) Pan angle of the camera with the reference coordinate system. (b) Tilt angle of the camera with the reference coordinate system.

$$\begin{pmatrix} ku \\ kv \\ k \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{f} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & f \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \quad (7)$$

Further, taking into consideration of the orientation of the camera described by the pan angle A , the tilt angle B shown in Figures 3(a) and (b) respectively and the skew angle C represents the orientation of the image plane centric system with respect to the camera coordinate systems. The above transformation takes the following form:

$$\begin{bmatrix} ku \\ kv \\ k \end{bmatrix} = T \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad (8)$$

where T is the perspective projection matrix given by

$$\begin{aligned}
T &= \begin{pmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{f} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & f \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos A & -\sin A & 0 \\ 0 & \sin A & \cos A & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \\
&\quad \times \begin{pmatrix} \cos B & 0 & \sin B & 0 \\ 0 & 1 & 0 & 0 \\ -\sin B & 0 & \cos B & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos C & -\sin C & 0 & 0 \\ \sin C & \cos C & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \\
&\quad \times \begin{pmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{9}
\end{aligned}$$

In the subsequent part of our discussion we will use the above expression for the computation of the perspective projection matrix T .

4. 3-D Reconstruction

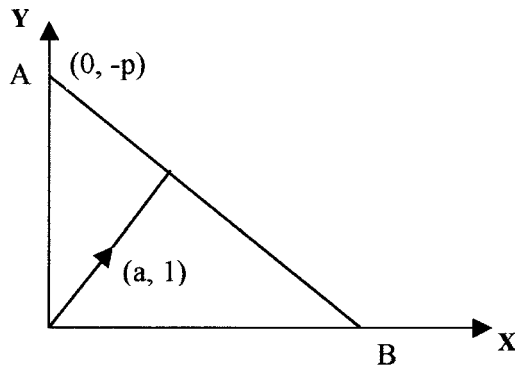
In this section we briefly outline the scheme for minimal representation [3] of 3-D points and 3-D planes and also demonstrate how EKF can be employed here for the reconstruction of the 3-D points from multiple 2-D points and the 3-D planes from multiple 3-D points on the plane. Let us first discuss briefly the possible representation of 2-D lines, 3-D lines, 3-D planes.

A 2-D line AB can be minimally best represented by two parameters a and p as evident from Figure 4. The advantage of this parameterization is that the equation of the lines are linear in the parameters (a, p) , which is essential in formulation of recursive Kalman filtering equation. Secondly the state vector which is derived from these parameters satisfy the inequality check criteria of the recursive Kalman filter. Similarly, the 3-D line CD can be minimally represented by four parameters a, b, p, q , as shown in Figure 5 and the plane EFGH can be represented by three parameters a, b, p as shown in Figure 6.

The parameters mentioned above help in representing the affine lines and planes in a minimum, complete and unambiguous manner, which can be used in subsequent estimation. Further, it is to be noted that the representation also satisfy the differentiability criteria to allow linearization of the measurement equation in the formulation of recursive Kalman filter, discussed earlier.

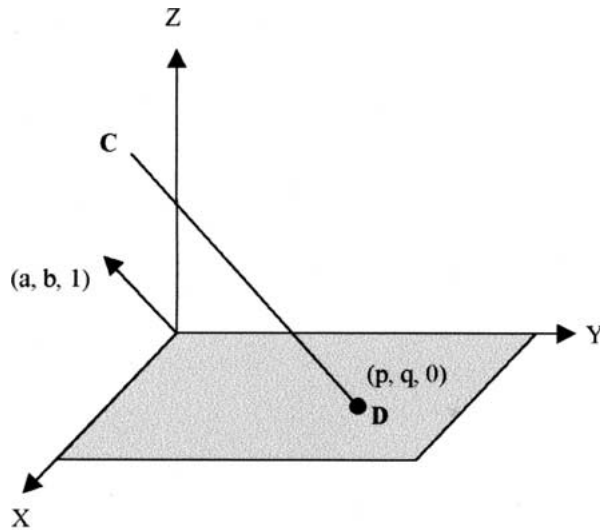
4.1. RECONSTRUCTION OF 3-D POINTS

Given the measured coordinates of the 2-D point (u_i, v_i) in several images, we can evaluate following Ayache's principle [3], the corresponding 3-D point using EKF.



$ax + y + p = 0$ (When the line not parallel to the Y-axis), or
 $x + ay + p = 0$ (When the lines are not parallel to the X-axis)

Figure 4. 2-D line AB that passes through $(0, -p)$ and normal to the line, passing through $(0, 0)$ and $(a, 1)$ can be represented by two parameter a and p .

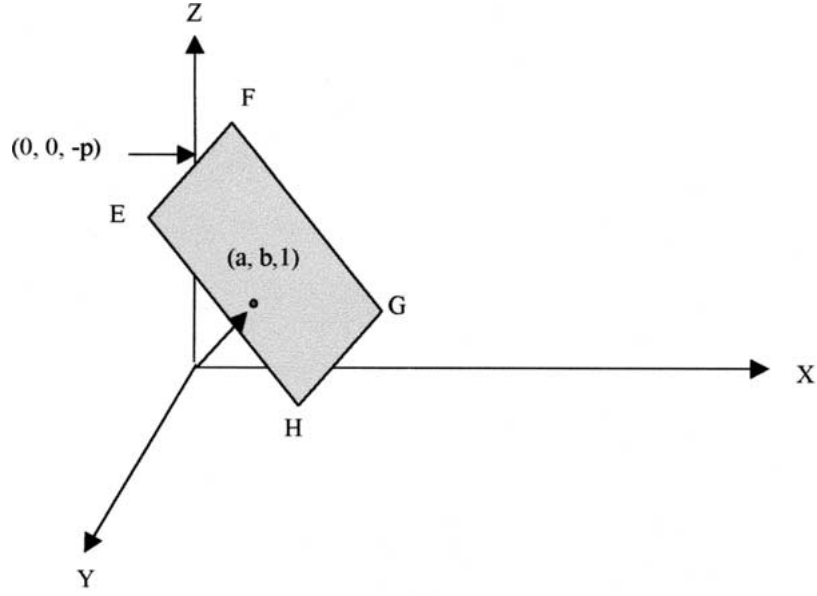


$x = az + p$ and $y = bz + q$ (when line not orthogonal to the Z axis)
 $y = ax + p$ and $z = bx + q$ (when line not orthogonal to the X axis)
 $z = ay + p$ and $x = by + q$ (when line not orthogonal to the Y axis)

Figure 5. A 3-D affine line CD that passes through the (X, Y) -plane at a point $(p, q, 0)$ and having the direction vector $(a, b, 1)^T$, can be represented by four parameters a, b, p, q .

Here M_i is a 2×3 matrix, y_i is a 2-dimensiona vector, and t_{ij} denotes the i th row and j th column element of the perspective matrix T (denoted by Equation (9)) and t_i is the first three element of i th row of the same matrix T . (u_i, v_i) is the 2-D image points of the vertex.

The measurement error matrix w_i is a 2×2 matrix estimated from the following expression



$$\begin{aligned}
 & \mathbf{a}x + \mathbf{b}y + z + p = 0 \text{ (when the planes not parallel to the Z axis)} \\
 & x + \mathbf{a}y + \mathbf{b}z + p = 0 \text{ (when the planes not parallel to the X axis)} \\
 & \mathbf{b}x + y + \mathbf{a}z + p = 0 \text{ (when the planes not parallel to the Y axis)}
 \end{aligned}$$

Figure 6. A 3-D affine plane EFGH that passes through $(0, 0, -p)$ and normal to the plane passing through $(0, 0, 0)$ and $(a, b, 1)$ can be represented by three parameters a, b, p .

$$\frac{\partial f}{\partial x_i} = \begin{pmatrix} -t_3 a_{i-1} - t_{34} & 0 \\ 0 & -t_3 a_{i-1} - t_{34} \end{pmatrix}. \quad (10)$$

Further, the estimator vector \mathbf{a} in the present context is the 3-D point $[x, y, z]^T$ with a initial value of $[0, 0, 0]^T$; \mathbf{S}_0 which is of 3×3 size in this case is initialized with a large value. Examples for the construction of 3-D points will be presented in detail in Section 5.

4.2. RECONSTRUCTION OF 3-D PLANE

For the reconstruction of the 3-D plane [3, 4], the filter has been employed with a different set of measurement and estimation vector. These vectors can be represented as follows:

$$\mathbf{a} = \text{estimation vector of the plane} = [a, b, p],$$

where a, b, p have the usual meaning from Figure 6, for the case when the plane is not parallel to Z-axis. The measurement vector (x_i) is the 3-D points (x, y, z) . We can choose the measurement equation as

$$f_i(x, a) = ax_i + by_i + z_i + p = 0. \quad (11)$$

After linearization the following expression can be derived for estimation,

$$y_i = -z_i, \quad (12)$$

$$M_i = [x_i \quad y_i \quad 1], \quad (13)$$

$$\frac{\partial f}{\partial x} = [a_{i-1} \quad b_{i-1} \quad 1]. \quad (14)$$

An example of reconstruction of 3-D planes from multiple 3-D points on the plane will be presented in Section 5.

5. Experimental Setup and Simulation

5.1. THE EXPERIMENTAL SETUP

The experiments were carried out on a Pioneer 2 mobile robot (Figure 7) manufactured by the ActivMedia Robotics company, USA. The system comprises of one tabletop client and a mobile robot acting as the server. The robot serves the client with its control programs for image grabbing, camera pan-tilt angle control and motion in prescribed directions. The client generates the command for the necessary tasks, and transmits them to the server agent (robot) through a radio communication link. The robot grabs images at prescribed camera position and transmits the TCP/IP packet of video bit streams for the tabletop client for subsequent image processing. The robot also makes its server aware of its current camera pinhole position as determined by it from the reading of its encoder disc connected with the wheels. It may be added here that the encoder disc provides the speed information, which is integrated by the agent to determine the camera pinhole position.

The robot in our experiment was commanded to move on a prescribed path (*vide* Figure 8) to take snaps of a wooden block (set 1 to set 4 images) at different camera orientations for the purpose of 3-D reconstruction.

5.2. OBJECT RECOGNITION BY EKF

An outline of the major steps involved in object recognition by EKF is presented below.

THE ALGORITHM FOR OBJECT RECOGNITION.

1. Take a new image at a given camera orientation with a predefined pinhole position.
2. Segment and localize the object of interest (for example, the wooden block in Figure 8) on the image.
3. Corresponding to the localized region on the original image find edges of the object by using Laplacian operator.

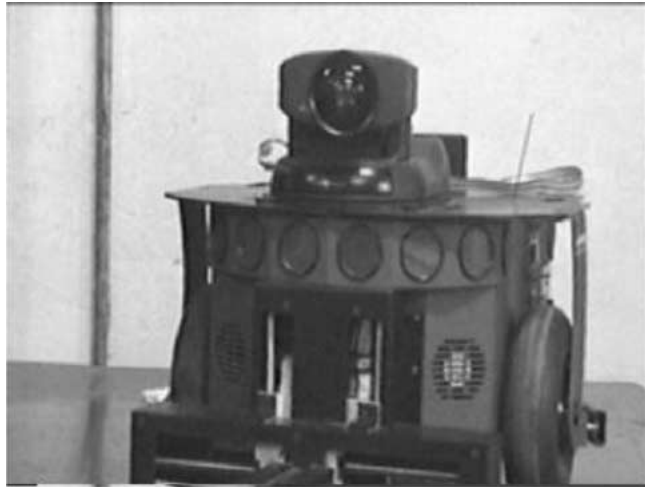


Figure 7. Photograph of the experimental Pioneer 2 mobile robot.

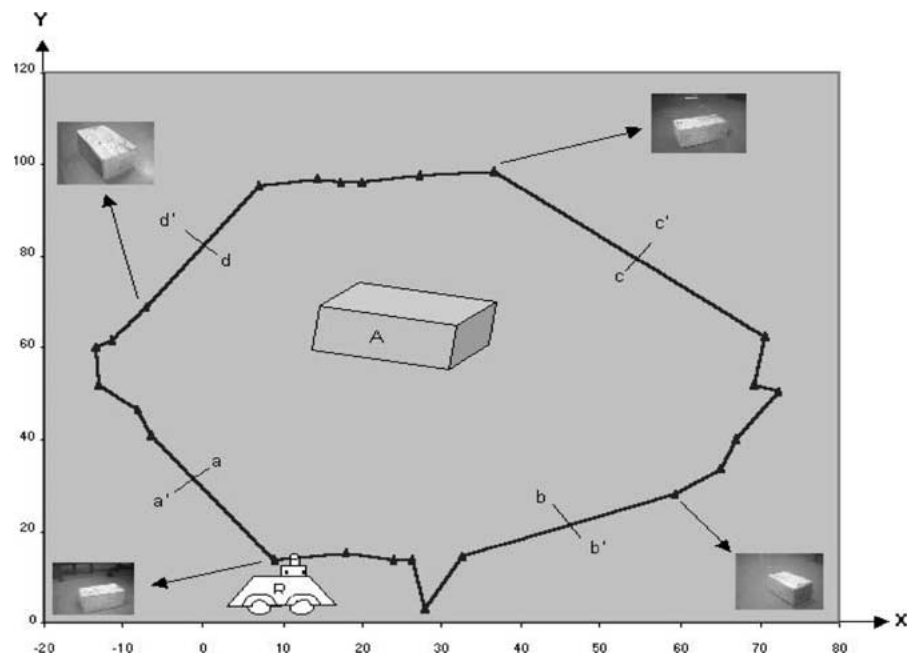


Figure 8. The trajectory of camera movement by a robot R around the block A to grab its image from 24 different locations, denoted by triangles. Grabbed images at few locations are shown.

SET-I



Visual Map_1
Pinhole Position: $x=9, y=14, z=28.5,$
 $A = 130^{\circ}, B=90^{\circ}$



Visual Map_2
Pinhole Position: $x=18, y= 15.5, z=28.5,$
 $A=0^{\circ}, B=90^{\circ}$



Visual Map_3
Pinhole Position: $x=24, y=13.5, z=28.5,$
 $A=0^{\circ} B=90^{\circ}$



Visual Map_4
Pinhole Position: $x=26.5, y=13.7, z=28.5,$
 $A = 13^{\circ}, B=90^{\circ}$



Visual Map_5
Pinhole Position: $x=28, y=3, z=28.5,$
 $A=0^{\circ} B=90^{\circ}$



Visual Map_6
Pinhole Position: $x=32.5, y=14.8, z=28.5,$
 $A=15^{\circ} B=90^{\circ}$

Figure 8. (Continued.)

SET-II



Visual Map_1
Pinhole Position: $x=59.5, y=28.2, z=28.5,$
 $A=53^{\circ}, B=130^{\circ}$



Visual Map_2
Pinhole Position: $x=65, y=34, z=28.5,$
 $A=45^{\circ}, B=90^{\circ}$



Visual Map_3
Pinhole Position: $x=67.0, y=40.0, z=28.5,$
 $A=71^{\circ}, B=90^{\circ}$



Visual Map_4
Pinhole Position: $x=72.3, y=50.1, z=28.5,$
 $A=78^{\circ}, B=90^{\circ}$



Visual Map_5
Pinhole Position: $x=69.5, y=52.0, z=28.5,$
 $A=90^{\circ}, B=90^{\circ}$



Visual Map_6
Pinhole Position: $x=70.7, y=62.5, z=28.5,$
 $A=100^{\circ}, B=90^{\circ}$

Figure 8. (Continued.)

SET-III



Visual Map_1
Pinhole Position: $x=36.5, y=98.5, z=28.5,$
 $A = 160^{\circ}, B=90^{\circ}$



Visual Map_2
Pinhole Position: $x=27.5, y=97.5, z=28.5,$
 $A = 180^{\circ}, B=90^{\circ}$



Visual Map_3
Pinhole Position: $x=20.0, y=96.0, z=28.5,$
 $A = 185^{\circ}, B=90^{\circ}$



Visual Map_4
Pinhole Position: $x=17.5, y=96.0, z=28.5,$
 $A = 190^{\circ}, B=90^{\circ}$



Visual Map_5
Pinhole Position: $x=14.5, y=96.5, z=28.5,$
 $A = 192^{\circ}, B=90^{\circ}$



Visual Map_6
Pinhole Position: $x=7.0, y=95.0, z=28.5,$
 $A = 200^{\circ}, B=90^{\circ}$

Figure 8. (Continued.)

SET-IV



Visual Map_1
Pinhole Position: $x=-7.0, y=69.0, z=27.0,$
 $A=242^{\circ}, B=140^{\circ}$



Visual Map_2
Pinhole Position: $x=-11.5, y=62.0, z=27.0,$
 $A=245^{\circ}, B=140^{\circ}$



Visual Map_3
Pinhole Position: $x=-13.5, y=59.7, z=28.5,$
 $A=270^{\circ}, B=90^{\circ}$



Visual Map_4
Pinhole Position: $x=-13.0, y=52.0, z=28.5,$
 $A=-80^{\circ}, B=90^{\circ}$



Visual Map_5
Pinhole Position: $x=-8.0, y=46.5, z=28.5,$
 $A=70^{\circ}, B=90^{\circ}$



Visual Map_6
Pinhole Position: $x=-6.5, y=40.8, z=28.5,$
 $A=65^{\circ}, B=90^{\circ}$

Figure 8. (Continued.)

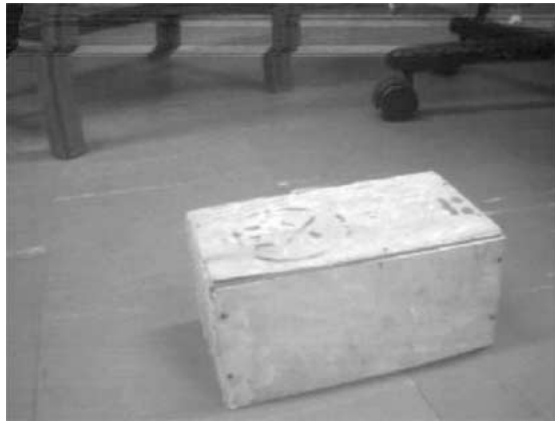
4. Find the 2-D vertices on the image by identifying a change in geometric slope of the pixels lying on the edges. If the change is significant of the order of 30 degrees or more (say), then those pixels are considered as 2-D vertices.
5. For each successive orientation of the camera and the pinhole position
 - Repeat steps 1–4;
 - Find the correspondence in the vertices of each 2 successive frames by the method presented in the next section;
 - Determine the 3-D points for each 2-D vertex of the object by taking the corresponding measurement of the 2-D vertex position in the next frame using EKF;
 - End For;
6. Mark the planes on the localized object in the original image by segmenting them further into different regions using average intensity of 8-neighbourhood points of a pixel as a measure. If more than one vertex is available in each segmented region then consider them belonging to the same plane.
7. Use EKF to determine the equation of planes using the corresponding 3-D points computed in step 5 corresponding to the 2-D points lying on a plane as found in step 6.
8. Determine the geometric relations such as perpendicularity or parallelism of the planes to identify the object of interest from its geometric definition.
9. End.

It is to be mentioned here that to keep the algorithm simple we omit the steps of robot movement and camera orientation control part from the algorithm. The camera orientation should be altered in successive frame and robot also needs to be moved a little to get the view of the most of the corresponding points in the next image. The successive positions of the robot and the camera orientations in our experiments were fixed from prior knowledge. A complete automation where the robot can itself decide its next position and camera orientation is under development.

5.3. EXPERIMENTAL DETAILS AND RESULTS

In step 2 of the algorithm we segmented the image to isolate the block from the background image. For segmentation the traditional thresholding technique was used to convert the gray image to a binary image. The pixels having gray value less than the selected threshold are assigned to have a gray value 0, while those exceeding or equal to the threshold are assigned a gray value 255. Since the block was white, a threshold of around 150 can isolate it from the rest of the gray image. A threshold less than 150 keeps part of the block dark, while a threshold more than this keeps regions close to the block white. Figure 9 shows one image taken by the robot and its corresponding segmented view.

After segmentation is over the white largest region corresponds to the block. Had there been a number of large white regions, then we need to use a window



(a)



(b)

Figure 9. (a) A snapshot of the box taken by the camera fixed with the robot. (b) The segmented figure (a) with a threshold = 150.

based localization scheme, where the entire image is partitioned into equal sized blocks and the blocks containing 80% or more white region is considered as the desired sub-block. Then the connected sub-blocks are identified and the location of the desired object is detected through shape analysis by the chain-code technique. Fortunately, the noisy white region in Figure 9(b) being small enough compared to the large sized white block may be ignored.

The localized block is then marked in the original image in Figure 9(a). The Laplacian operator then is used to determine the edges of the block (Figure 10). For the detection of the vertices, the pixels on the edges are traced and at each pixel the change in geometric direction gradient is determined. The pixels having a change in slope greater than 30 degrees are regarded as the 2-D vertices. It may be noted that one vertex has been lost in image 9(b), but this can be avoided if

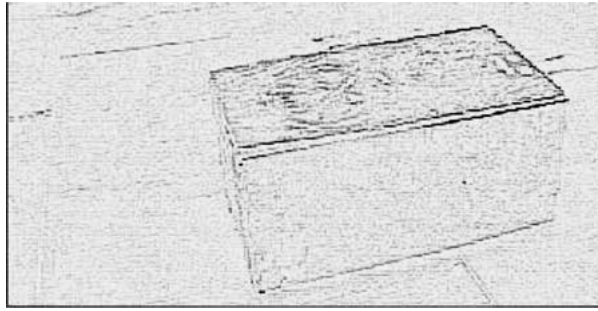


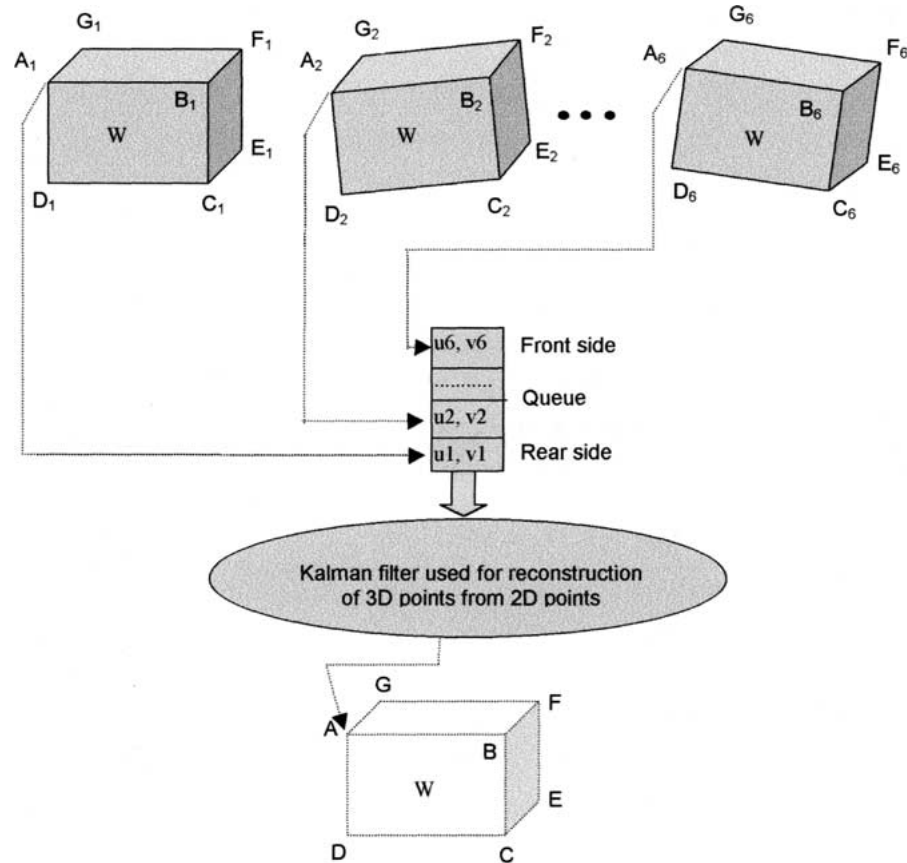
Figure 10. The edges of the block in image 9(a) after localization.

region growing technique was employed for segmentation. But region growing is highly time-consuming, so we used histogram-based thresholding technique for segmentation.

Once the 2-D vertices of an object in an image are identified they are labeled as back-left-top, back-right-top, front-right-top, front-left-top, front-left-bottom, front-right-bottom, etc. depending on their positions in the block. This is actually needed to handle the **correspondence problem**. The correspondence problem is concerned with identifying the same points in 2 or more images. After labeling the vertices in the image in that manner identifying the same vertex in 2 images becomes easier. In case more than one vertex in an image has the same label, say back-left-top, or there exist possibilities that two or more vertex can have the same labels then additional junction labeling scheme needs to be adopted. The junction-labeling scheme by Waltz [19] that labels the junctions into 18 possible labels, for instance is a good choice. If more than one type of Waltz's 18 junction labels is available in a scene, then a combination of our labeling scheme and Waltz's junction labeling scheme may be adopted. We in our experiment, however, have used the position-wise labels to the vertices. For determining the type of junction labels, the following type rules have been employed.

RULE 1: *If (out of the neighboring 8 pixels only the south, east and south-east pixels have identical gray value, and the rest are different)
Then (the vertex is **back – left – top** corner).*

Once the correspondence in the 2-D vertices of the block in multiple images is detected, they can be supplied one by one to the EKF for the determination of their corresponding 3-D point. As the robot moves around the obstacle (Figure 8) it takes 6 snaps of the block W between the line segments aa' to bb' , bb' to cc' , cc' to dd' and dd' to aa' . Coordinates of the 2-D vertices of the block W , measured from these six images between 2 line segments (say aa' to bb'), have been used recursively as the input to a Kalman filter for the reconstruction of their 3-D coordinates. Figure 11 illustrates the input and output parameters of a Kalman filter employed for reconstruction of 3-D coordinates of a vertex 'A' from six 2-D image coordinate ' A_1 ' through ' A_6 ' of same vertex 'A'.



2D points A_1 through A_6 have been fused to point A by Kalman filter. The other points B, C, D, E, F, F in the figure have been reconstructed similarly.

Figure 11. Extraction of the 3-D points from six 2-D image points.

The first two iterations of the estimation process of reconstructing the 3-D point from the multiple 2-D image points are as follows. The inputs to the first iteration are 2-D point $(u_1, v_1) = (-2.7, 1.3)$ and set of camera parameters $x_0 = 36.5, y_0 = 98.5, z_0 = 28.5, A = 2.793, B = -1.97, C = 0$. The output is the 3-D point $(x, y, z) = (22.11, 6.84, -19.57)$ and the covariance matrix = $[232.6 \ 1192.9 \ 627.5; \ 1192.9 \ 7699.3 \ 4021.5; \ 627.5 \ 4021.5 \ 2161.7]$. Similarly the input to the second iteration are the 2-D point $(u_2, v_2) = (-2.25, 1.3)$ and a new set of camera parameters $x_0 = 27.5, y_0 = 97.5, z_0 = 28.5, A = 3.14, B = -1.97, C = 0$. The output of is the 3-D point $(x, y, z) = (32.04, 63.64, 10.44)$ and the covariance matrix is $[22.79 \ -8.92 \ -4.42; \ -8.92 \ 805.88 \ 413.28; \ -4.43 \ 413.28 \ 240.84]$. The procedure continues until all the scanned points taken by the camera are exhausted or the covariance matrix is very close to a null matrix.

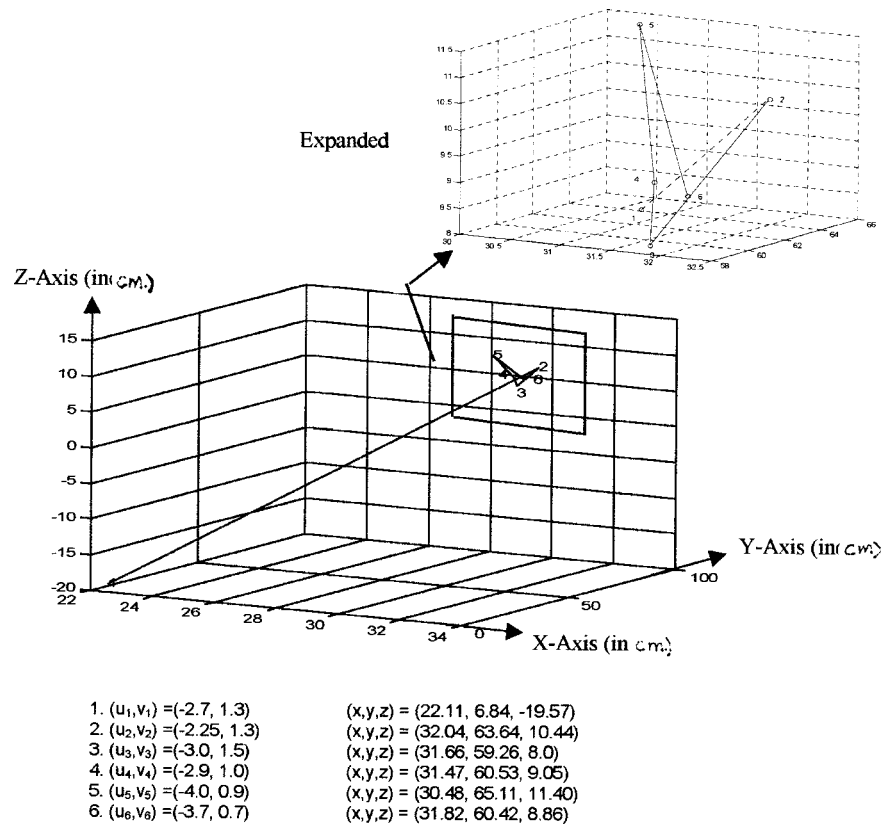


Figure 12. The 3-D reconstruction points with incoming stream of 2-D points A_1-A_6 .

The response of the Kalman filter in the reconstruction of the 3-D points is presented in Figure 12 and one element of the covariance matrix against iterations is presented in Figure 13. It is clear from these figures that a higher accuracy in the 3-D reconstruction can be achieved with increased number of 2-D image points.

After reconstruction of the 3-D vertices of a planer object the location of the planes in the object need to be identified. This has been carried out by another stage of segmentation with multiple thresholding. The frequency histogram for the wooden block has multiple peaks. The image needs to be converted to a binary image after setting a threshold marginally exceeding the gray value corresponding to the peak of the histogram. Most of the planes can be located by repeating this process for each peak of the histogram. Next important aspect is to identify the vertices that lie on each plane. Since the vertices are already identified and labeled, this step is rudimentary. The 3-D planes now can be reconstructed with the knowledge of the 3-D points on the plane. An example below illustrates this.

Suppose, one 3-D vertex is given by $(x, y, z) = (0, 0, 3)$ along with a initial measure of \mathbf{a}_0 (= null vector) and \mathbf{S}_0 (= a large matrix). The EKF at its first iteration then estimates the 3-D plane parameters $(a, b, p) = (0, 0, -2.997)$. At the

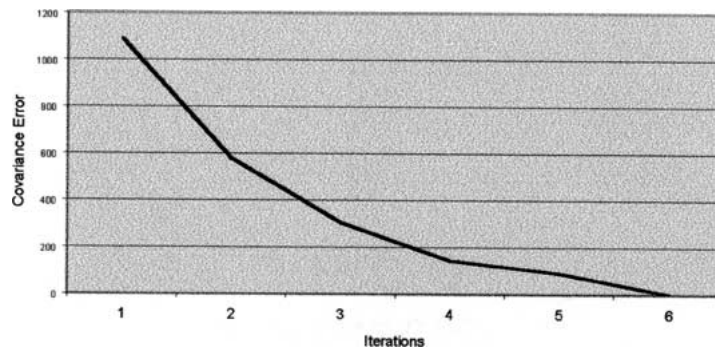


Figure 13. One element of the error covariance matrix S_j .

second iteration the input of the filter is the 3-D point $(x, y, z) = (3.2, 2.2, 3.1)$ along with \mathbf{a}_1 and S_1 estimated in the first iteration and the output of the filter is the 3-D plane parameter $(a, b, p) = (-0.022, -0.015, -2.997)$. The procedure is thus repeated with number of points on the planes until the covariance matrix is very close to a null matrix or all 3-D points available on the plane have been exhausted.

The final part of the problem is to determine the 3-D planer object from the relationship of its 3-D planes. To illustrate, let us consider a physical object, say the box, shown in Figure 14. Let us define a box as an object consisting of at least three planes abcd, cdef and adeg having an angle α , β , and δ between them, where the angles can take the values $90^\circ - \theta \leq \alpha, \beta, \delta \leq 90^\circ + \theta$, and $(0 \leq \theta \leq 15^\circ)$. The following types of rule have been employed to identify planer objects like box, chairs or tables.

IF (Angle-between (plane 1, plane 2, $90^\circ + \theta$) AND
 Greater-than ($\theta, 0^\circ$) AND Less-than ($\theta, 15^\circ$) AND
 Angle-between (plane 2, plane 3, $90^\circ + \theta$) AND
 Greater-than ($\theta, 0^\circ$) AND Less-than ($\theta, 15^\circ$) AND
 Angle-between (plane 3, plane 1, $90^\circ + \theta$) AND
 Greater-than ($\theta, 0^\circ$) AND Less-than ($\theta, 15^\circ$))
 THEN (object = box).

In our implementation we realized this part by logic programming [1] using PROLOG since it can handle logical relations very efficiently.

To take care of the partial matching of the object features with those of model objects we are now developing a fuzzy-logic-based matching system [5]. A fuzzy membership function is defined to describe the object feature in membership domain and a partial matching between the object features and the geometric primitives of an ideal object is carried out in this domain. Depending on the degree of matching, the object is classified to one of the model objects.

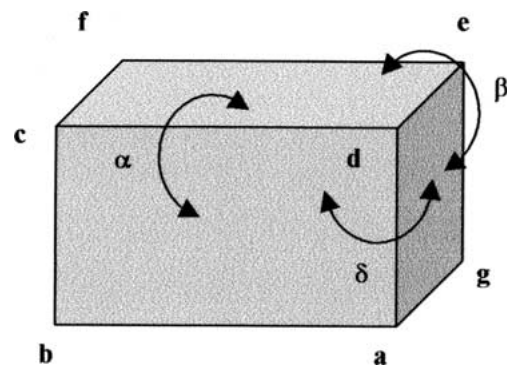


Figure 14. Spatial relations among components of a box.

6. Conclusions

The paper presented an experimental scheme for 3-D reconstruction of planar objects from their multiple images using extended Kalman filtering. The proposed technique has been applied in recognizing simple objects like box in a robot's workspace. The well-known 3-D map-building problem in mobile robotics thus can be realized by the proposed approach. For the recognition of a 3-D object the robot needs to move around the object and take multiple snaps at different camera orientations. The time required for grabbing an image, transferring it to the tabletop client and image processing is around 200 milliseconds. The major time required is to move the robot around the block. The time required to recognize a wooden block of (10" \times 8" \times 6") dimension taking into account the motion of the robot has been found to be 18 seconds approximately on a Pentium III type client with 64 MB RAM and the Pioneer 2 type server.

References

1. Alferes, J. J. and Pereira L. M.: *Reasoning with Logic Programming*, Springer, Heidelberg, 1996.
2. Asada, M.: Map building for a mobile robot from sensory data, *IEEE Trans. Systems Man Cybernet.* **37**(6) (1990), 1326–1336.
3. Ayache, N.: *Artificial Vision for Mobile Robot*, MIT Press, Cambridge, MA, 1991.
4. Ayache, N. and Faugeras, O. D.: Building a consistent 3-D representation of the mobile robot environment by combining multi stereo views, in: *Proc. of the Internat. Joint Conf. on Artificial Intelligence*, August 1987.
5. Biswas, B., Konar, A., and Mukherjee, A. K.: Fuzzy moments for digital image matching, in: *Proc. of Internat. Conf. on Control, Automation, Robotics and Computer Vision, ICARCV '98*, 1998; also communicated to *Engineering Applications of Artificial Intelligence*, Elsevier Publications, North-Holland.
6. Brown, R. G. and Hwang, P. Y. C.: *Introduction to Random Signals and Applied Kalman Filtering*, Wiley, New York, 1997.
7. Dean, T., Allen, J., and Aloimonos, Y.: *Artificial Intelligence: Theory and Practice*, Addition-Wesley, Reading, MA, 1995.

8. Elfes, A.: Sonar-based real-world mapping and navigation, *J. Robotics Automat.* **3**(3) (1987), 249–264.
9. Jahne, B.: *Practical Handbook on: Image Processing for Scientific Applications*, CRC Press, Boca Raton, FL, 1997.
10. Kalman, R. E.: A new approach to linear filtering and prediction problems, *Trans. ASME J. Basic Engrg.* (March 1960), 35–45.
11. Merek, V. W., Nerode, A., and Truszczyński, M.: *Logic Programming and Non Monotonic Reasoning, Lecture Notes in Artificial Intelligence*, Springer, Berlin, 1995.
12. Pagac, D., Nebot, E. M., and Durrant-Whyte, H.: An evidential approach to map-building for autonomous vehicles, *IEEE Trans. Robotics Automat.* **14**(4) (1998), 623–629.
13. Patnaik, S.: Building cognition for mobile robots, PhD Thesis, Submitted in Jadavpur University, September 1999.
14. Patnaik, S., Konar, A., and Mandal, A. K.: Map building and navigation by a robotic manipulator, in: *Proc. of Internat. Conf. on Information Technology*, Bhubaneswar, December 1998, TATA/McGraw-Hill, pp. 227–232.
15. Patnaik, S., Konar, A., and Mandal, A. K.: Visual perception for navigational planning and coordination of mobile robots, *Indian J. Engrg.* **26** (1998), 21–37.
16. Patnaik, S., Konar, A., and Mandal, A. K.: Building 3-D visual perception of a mobile robot using extended Kalman filtering, in: *Proc. of the All India Seminar on Information Technology: Opportunity and Challenges*, March 2000, pp. 58–63.
17. Schalkoff, R. J.: *Digital Image Processing and Computer Vision*, Wiley, New York, 1989.
18. Taylor, C. J. and Kriegman, D. J.: Vision-based motion planning and exploration algorithm for mobile robots, *IEEE Trans. Robotics Automat.* **14**(3) (1998), 417–426.
19. Waltz, D.: Understanding line drawings of scenes with shadows, in: P. H. Winston (ed.), *Psychology of Computer Vision*, MIT Press, Cambridge, MA, 1972.