Tracking Line Segments *

Rachid DERICHE - Olivier FAUGERAS
INRIA Sophia-Antipolis
2004 Route des Lucioles
06565 Valbonne Cedex France
deriche@rhodes.inria.fr

Abstract

This paper describes the development and the implementation of a line segments based token tracker. Given a sequence of time-varying images, the goal is to track line segments corresponding to the edges extracted from the image being analyzed. We will present a tracking approach that combines a prediction and a matching steps. The prediction step is a Kalman filtering based approach that is used in order to provide reasonable estimates of the region where the matching process has to seek for a possible match between tokens. Correspondence in the search area is done through the use of a similarity function based on *Mahalanobis* distance between attributs carefully chosen of the line segments. The efficiency of the proposed approach will be illustrated in several experiments that have been carried out considering noisy synthetic data and real scenes obtained from the INRIA mobile robot.

1 Introduction

The problem we want to adress is to detect and track features in a sequence of time varying images acquired by a camera on a robot moving in an indoor environment. At this end, edges are first detected through the use of an optimal operator developed in one of our previous work (see [1]). An edge linking step and a polygonal approximation yield the line segments that best approximate the extracted edges. A tracking approach based on a close cooperation between a prediction and a matching step is then developed. The prediction step utilizes a Kalman filter to estimate the position in the next image of each primitive in the current image. The matching step then need to be applied only within the predicted region. Matching employs a normalized distance based on the most stable parameters of each primitive.

The organization of the paper is as follows : A first section presents the problem of the representation for the line segments, and shows in particular why the mid point representation has been chosen. Section 2 gives a general overview of the developed approach and presents how the algorithm works. Section 3 explains in particular why

*This work was partially completed under Esprit P940

adding an error to the model is strongly recommended in our approach and how this remark can be taken into account in the Kalman filtering equations. Section 4 is devoted to the presentation of the matching process. A last section, before the conclusion, is then devoted to the experiments carried out on synthetic and real data.

2 What representation for the line segments ?

An important sub problem is to choose an appropriate representation for the line segments since the tracking will be based on this representation . It is clear for example that tracking both endpoints of each segment will be very difficult, since they are not at all reliable due to the fact that segments can be broken from one frame to another. For this reason, two types of representations have been considered.

2.1 Two Representations for Line Segments

In the first representation, a line segment having endpoints at points $P1(x_1, y_1)$ and $P2(x_2, y_2)$ is characterized by the vector $\mathbf{v_1} = [c, d, \theta, l]^T$ where θ represent the orientation of the line segment, l its length, the parameter c denotes the distance of the origin to the line segment and the parameter d, the distance along the line from the perpendicular intersection to the midpoint of the segment. These parameters are derived from the endpoints as follows :

$$\begin{cases} \theta = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \\ l = \sqrt{((x_2 - x_1)^2 + (y_2 - y_1)^2)} \\ c = \frac{(x_2 * y_1 - x_1 * y_2)}{l} \\ d = \frac{(x_2 - x_1) * (x_1 + x_2) + (y_2 - y_1) * (y_2 + y_1)}{2 * l} \end{cases} \tag{1}$$

The second representation that we considere is the midpoint representation and characterizes a line segment by the vector $\mathbf{v_2} = [x_m, y_m, \theta, l]^T$ where the point (x_m, y_m) defines the coordinates of the mid point Pm of the segment. Figure 1 illustrates both representations.

2.2 c,d,θ,l or midpoint representation ?

In order to find what representation is more appropriate for our tracking algorithm, we have to calculate the covariance matrices associated to the vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ that define both representations from those of the endpoints. At this end we use the nonlinear relations that give theses vectors function of the vector $\mathbf{p}=[x_1, y_1, x_2, y_2]^T$ and the following relation :

$$\Lambda_{\mathbf{v}} = \frac{\partial \mathbf{v}}{\partial \mathbf{p}} \Sigma \frac{\partial \mathbf{v}}{\partial \mathbf{p}}^T \tag{2}$$

Where $\frac{\partial \mathbf{v}}{\partial \mathbf{p}}$ is a the 4*4 Jacobian matrix and $\mathbf{\Sigma}$ is the 4*4 covariance matrix of the vector $\mathbf{p}=[x_1, y_1, x_2, y_2]^T$. Assuming no correlation between the endpoints and the same covariance matrix $\mathbf{\Lambda}$ for both endpoints leads to the following covariance matrix $\mathbf{\Sigma}$:

$$\mathbf{\Sigma} = \begin{pmatrix} \mathbf{\Lambda} & 0 \\ 0 & \mathbf{\Lambda} \end{pmatrix} \tag{3}$$

where $\mathbf{\Lambda}$ is the 2*2 covariance matrix associated to each endpoint.

$$\mathbf{\Lambda} = \begin{pmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_y^2 \end{pmatrix} \tag{4}$$

Due to the fact that line segments may be broken differently from one image to another, an endpoint is not reliable. We model this segmentation noise, introduced by the polygonal approximation, as follows : We assume that $\mathbf{\Lambda}$ is diagonal in the coordinate system defined by $\mathbf{u}_{\parallel}$ and $\mathbf{u}_{\perp}$, two units vectors parallel and perpendicular to the line segment, respectively. In this coordinate system, $\mathbf{\Lambda}$ is written :

$$\mathbf{\Lambda}_{\perp\parallel} = \begin{pmatrix} \sigma_{\parallel}^2 & 0 \\ 0 & \sigma_{\perp}^2 \end{pmatrix} \tag{5}$$

Noting that the coordinate system defined by $\mathbf{u}_{\parallel}$ and $\mathbf{u}_{\perp}$ is obtained using a rotation of an angle θ around the origin, leads to the following relations for the covariance matrix $\mathbf{\Lambda}$:

$$\begin{cases} \sigma_x^2 = \sigma_{\parallel}^2 * cos(\theta)^2 + \sigma_{\perp}^2 * sin(\theta)^2 \\ \sigma_y^2 = \sigma_{\perp}^2 * cos(\theta)^2 + \sigma_{\parallel}^2 * sin(\theta)^2 \\ \sigma_{xy}^2 = (\sigma_{\parallel}^2 - \sigma_{\perp}^2) * sin(\theta) * cos(\theta) \end{cases} \tag{6}$$

Applying these results to the vectors $\mathbf{v_1} = [c, d, \theta, l]^T$ and $\mathbf{v_2} = [x_m, y_m, \theta, l]^T$, we find after some algebra that their covariance matrices $\mathbf{\Lambda_{v_1}}$ and $\mathbf{\Lambda_{v_2}}$ respectively are given as follows :

$$\mathbf{\Lambda_{v_1}} = \begin{pmatrix} \frac{\sigma_{\perp}^2}{2} + \frac{2*d^2*\sigma_{\perp}^2}{l^2} & \frac{-2*c*d*\sigma_{\perp}^2}{l^2} & \frac{-2*d*\sigma_{\perp}^2}{l^2} & 0 \\ \frac{-2*c*d*\sigma_{\perp}^2}{l^2} & \frac{\sigma_{\parallel}^2}{2} + \frac{2*c^2*\sigma_{\perp}^2}{l^2} & \frac{2*c*\sigma_{\perp}^2}{l^2} & 0 \\ \frac{-2*d*\sigma_{\perp}^2}{l^2} & \frac{2*c*\sigma_{\perp}^2}{l^2} & \frac{2*\sigma_{\perp}^2}{l^2} & 0 \\ 0 & 0 & 0 & 2*\sigma_{\parallel}^2 \end{pmatrix} \tag{7}$$

$$\mathbf{\Lambda_{v_2}} = \begin{pmatrix} \frac{\sigma_{\parallel}^2*cos(\theta)^2+\sigma_{\perp}^2*sin(\theta)^2}{2} & \frac{(\sigma_{\parallel}^2-\sigma_{\perp}^2)*sin(\theta)*cos(\theta)}{2} & 0 & 0 \\ \frac{(\sigma_{\parallel}^2-\sigma_{\perp}^2)*sin(\theta)*cos(\theta)}{2} & \frac{\sigma_{\perp}^2*cos(\theta)^2+\sigma_{\parallel}^2*sin(\theta)^2}{2} & 0 & 0 \\ 0 & 0 & \frac{2*\sigma_{\perp}^2}{l^2} & 0 \\ 0 & 0 & 0 & 2*\sigma_{\parallel}^2 \end{pmatrix} \tag{8}$$

From these results, one can say that the c, d, θ, l representation leads to a covariance matrix that depends strongly on the position of the associated line segment into the image through the parameters c and d that appear on the covariance matrix $\mathbf{\Lambda_{v_1}}$ Therefore two given segments with the same length and orientation will have their uncertainty on the (c, d) parameters completely differents depending on their position within the image. This is not the case for the mid point representation since the uncertainty associated to the mid point (x_m, y_m) depends only on the uncertainty of the endpoints.

A second important point to note for the mid-point representation is the decorrelation that exists between the parameters (x_m, y_m) and the parameters θ and l. This decorellation between the parameters does not exist for the representation c, d, θ, l. Adding to that, it is easy to check that for the particular case where $\sigma^2 = \sigma_\perp^2 = \sigma_\parallel^2$, then the four parameters x_m, y_m, θ, l are completely decorellated, while it is not the case for the c, d, θ, l representation, where we have to assume that $\sigma_\perp$ is equal to zero in order to decorellate the paramaters. This question of decorellation is important if we want to use for efficiency consideration, differents Kalman filters on each parameters.

From these remarks, it is clear that the mid point representation (x_m, y_m, θ, l) is more appropriate to our tracking algorithm where each segment is represented by four points in a 1 dimensional space : the x and y-position information and the length and orientation information. When a given segment moves in the image, these four points follow a trajectory in the 1D space. The kinematics of the motion of the given line segment is the kinematics of the fourth points i.e trajectory, velocity and acceleration. Therefore we will run four Kalman filters independently on each parameter.

3 Tracking

The tracking approach we have developed is based on the following steps :

1. Assign a kinematics model to each parameter of the representation of the line segment.
2. Use the model to predict the position of the given parameter in the next frame and the associated uncertainty.
3. Use the uncertainty to determine the search area around the predicted position.
4. Inside the search area, use a normalized distance to determine the best match.
5. If a match is found, use it as a new measure to update the kinematics model.

Initially a $t = 0$, no information is available about the kinematics of the line segments in the image. This is called the *bootstrapping stage.* We assume zero velocities and accelerations with large uncertainties indicating that we do not trust these guesses. We say that a line segment of the current frame is inside a searching area if the distance of each parameter to its predicted position is less than a given threshold. A similarity function that combines the fourth distances is then used in order to compute a score for each correspondance between the predicted position and the position of the line segments lying inside the searching area. Once a line segment from the current frame has been matched to a line segment of the model, its parameters are then used as a new measure in order to update the kinematics of the model.

A Kalman filter is used to perform tracking by providing reasonable estimates of the region where the matching process has to seek for a possible match between tokens. Kalman filtering is a statistical approach to estimate a time-varying state vector X_t from noisy measurements Z_t. Consider the estimation of X_{t+k} from the measurements up to the instant k, Kalman filtering is a recursive estimation scheme designed to match the

dynamic system model, the statistics of the error between the model and reality, and the uncertainty associated with the measurements.

In our application, four Kalman filters are applied independently on each parameter of the representation (x_m, y_m, θ, l). Each state vector is just the position of the given parameter x (i.e x_m, y_m, θ, l) , its velocity $\dot{x}$ and its acceleration $\ddot{x}$. The following discrete time steps notation is used for the state vector at the t^{th} time step $X_t = [x_t, \dot{x}_t, \ddot{x}_t]^T$. The model of the system dynamics and the measurements model are given as follows in our application :

$$\begin{cases} X_{t+1} = \Phi_{t+1,t} X_t + \xi_t \\ \\ V_t = C_t X_t + \eta_t \end{cases} \tag{9}$$

where ξ_k is assumed to be a zero mean Gaussian noise sequence of covariance Q_t representing the error of the model, $\Phi_{t+1,t}$ is the matrix which evolves the position x, the velocity $\dot{x}$ and the acceleration $\ddot{x}$ from one time sample to another. V_t is a vector of measurements with an uncertainty η_t, assumed to be a zero-mean Gaussian noise sequence of covariance R_t. In our application, $\Phi_{t+1,t}$ assumes a motion with constant acceleration and the measurement model assumes that the position x is measurable from the matching process while the velocity $\dot{x}$ and the acceleration $\ddot{x}$ are not.

The classical Kalman filtering equations (see [3] for example) allows to compute the optimal estimates of the state vector $\hat{X}_t = \hat{X}_{t/t}$ of X_t recursively from the data $V_0, V_1, ..., V_t$ and the initial estimation $E(X_0)$ and $Var(X_0)$.

4 Error Modelling

Implementing the classical theoretical Kalman filter equations leads to results strongly linked to the accuracy of the model. When there is a discrepancy between the true system model and the model assumed by the Kalman filter, the resulting estimation simply does not correspond to that predicted by theory. The Kalman filtering approach developed for our tracking algorithm is based on an assumed model of the trajectory (i.e constant velocity or acceleration). It is clear that this model can be considered as correct only *locally*. This means that we assume that the trajectory of each parameter of the line segment can always be approximated *locally* by a first or second degree polynomial. This is a more reasonable assumption than the one that considers that all the trajectory can be fitted by a first or second degree polynomial. Among all the solutions that have been proposed in the literature (see reference [3]) in order to deal with this discrepancy problem, we have chosen and implemented the following two techniques :

- **Add process noise.**

  This solution, consisting in the addition of process noise to the system model as done in equation 9 with the term ξ_t, is attractive for preventing divergence. In our application, the covariance Q_t of ξ_k in 9 is taken as :

$$\mathbf{E}(\xi_t \xi_t^{\mathbf{T}}) = \begin{pmatrix} \sigma_p^2 & 0 & 0 \\ 0 & \sigma_v^2 & 0 \\ 0 & 0 & \sigma_a^2 \end{pmatrix} \tag{10}$$

It can be shown that in such case, the Ricatti equation associated to the system model 9 can be resolved to get the following *steady state* Kalman filtering equations :

$$\hat{X}_{t|t} = \hat{X}_{t|t-1} + G(V_t - C_t\hat{X}_{t|t-1}) \tag{11}$$

Thus the filter will always track the data. This has to be compared to the case where no errors on the model were assumed. In such case the Kalman gain G_t will tend to zero as t tends to the infinity and thus the filter will no longer continue to track. The elements of Q (the values of σ_p^2,σ_v^2 and σ_a^2) depends to a great extent upon what is known about the unmodeled states. In our application all the used values have been derived from the simulation experiments. However the choice of σ_m^2, the variance on the position, is done in a manner reflecting our a priori estimate of the amount of noise to be expected from the previous step (Digitizing effects, edge detection and polygonal approximation).

It should be pointed out that the Ricatti equation may be solved before the filtering process is being performed. Therefore if we assign to the Kalman Gain G the following constant values $G = [\alpha, \beta]^T$ in our application that deals with a constant velocity model, then it can be shown that we obtain the following decoupled equations :

$$\begin{cases} x_{t/t} = -(\alpha+\beta-2) * x_{t-1/t-1} - (1-\alpha) * x_{t-2/t-2} + \alpha * v_t + (-\alpha+\beta) * v_{t-1} \\ \dot{x}_t = -(\alpha+\beta-2) * \dot{x}_{t-1/t-1} - (1-\alpha)\dot{x}_{t-2/t-2} + \beta * (v_t - v_{t-1}) \end{cases} \tag{12}$$

and the following covariance matrix $\mathbf{P}$:

$$\mathbf{P} = \frac{\sigma_m^2}{1-\alpha} \begin{pmatrix} \alpha & \beta \\ \beta & \beta * (\alpha+\beta) \end{pmatrix} \tag{13}$$

α is a real positive scalar less than 1 and β a real positive scalar less than $\frac{\alpha^2}{1-\alpha}$.

This limiting Kalman filtering is the well known near optimal α,β tracker [3]. It has been implemented in our application and found that it is extremely efficient since all the coefficients of the recursive equations given above can be calculated before the tracking starts.

- **Elimination of old data**

The basic idea in this approach is to consider the old data as no more meaningful and therefore to discard them. A simple way to accomplish this elimination is then to weight the old data according to when they occured. This means that the covariance of the measurement noise must somehow be increased for past measurement. It can be shown that one simple manner of accomplishing this is to multiply at each new measure the covariance prediction for the state vector with the age weighting scalar factor α greater than or equal to one [3].

5 Matching

The search area is determined through a simple set of attribute tests using the result of the Kalman filtering. For each token of the image flow model, represented by a feature vector of 4 components, we wish to know which token might correspond to it. In selecting a cost function for correspondences, we wanted to take into account the distance between the expected parameter value with its uncertainty and the current value of the measure with its uncertainty. This leads to calculate the so called *Mahalanobis* distance, for each components and to declare a token of the new frame inside a search area if all theses distances are less than a fixed threshold. Inside the search area, the correspondence is then controlled by the value of the sum of these distances. It is calculated for each possible match and the best score is used to validate the most consistent.

Let each new token, issued from the matching process, be represented by a feature vector of N components denoted T_m with a covariance matrix Γ. Let the estimated token represented by T_e with a covariance matrix Λ. It is then easy to find that in the case where no correlation exists between both vectors T_m and T_p, then the covariance matrix S of the vector difference $V = T_m - T_e$ is just the sum of Γ and Λ. The *Mahalanobis* distance is then defined to be:

$$d_{\chi^2} = (T_m, T_e) = (T_m - T_e)^T (S)^{-1} (T_m - T_e) \tag{14}$$

This distance has a χ^2 distribution with N degrees of freedom, where N=1 in our case. This distance deliminates the upper bound on the variation of V from its mean. The probability that d_{χ^2} is less than a given threshold ϵ_{χ^2} may be obtained from a χ^2 distribution table. In order to deal with a search area where we have a probability of 95% to find the measure, we set the value of the threshold to 3.84.

6 Experimental Results

Many experiments have been carried out considering several noisy synthetic data and differents real scenes, however due to the limited number of pages, this part have benn considerably reduced from its original version. Figure 2 illustrates a noisy trajectory that have been synthetized in order to simulate an inaccurate constant+ramp model and the results given by the prediction without error modelling. Note that the results have not been superimposed but displaced in position in order to better compare the two trajectories. Figure 3 shows the predicted trajectories taking into account an error modelling through the value assigned to σ_v^2 (0.01 and 0.1). It appears that the lower the variance σ_v^2 is, the more serious the divergence problem is. On the other hand, the higher the variance is, more noisy the prediction is, because the filter takes into account few data.

The tracking on real data is illustrated in Figures 4 and 5 through the numbers assigned to each line segment. A close look at the results reveals how some line segments can appear or disappear. A new label is affected as soon as a new segment appears and the process continues without affecting the tracking algorithm. Through the number of real experiments, it has been demonstrated that the approach developped works well and gives satisfatory results. This approach is now an integral part of the standard project

demonstration and our industrial partner ITMI is incorporating it into hardware for the DMA machine of the Esprit project P940.

7 Conclusion

A line segments based token tracker has been developed and implemented. Given a sequence of time-varying images, it allows to track line segments corresponding to the edges extracted from the scene being analyzed. The results obtained through many experiments on real data seems very promising. We are currently working on the exploitation of these results for computing 3D motion and structure [2].

References

[1] R.Deriche. Using canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2):167–187, May 1987.

[2] O.D.Faugeras. R.Deriche. N.Navab From Optical Flow of Lines to 3D Motion and structure. *Proceedings IEEE Int. Work. on Intell. Syst.* pp 646-649, Sept 1989. Tsukuba, Japan.

[3] A.Gelb and al Applied Optimal Estimation *The Analytic Sciences Corporation* ed. Arthur Gelb. M.I.T Press

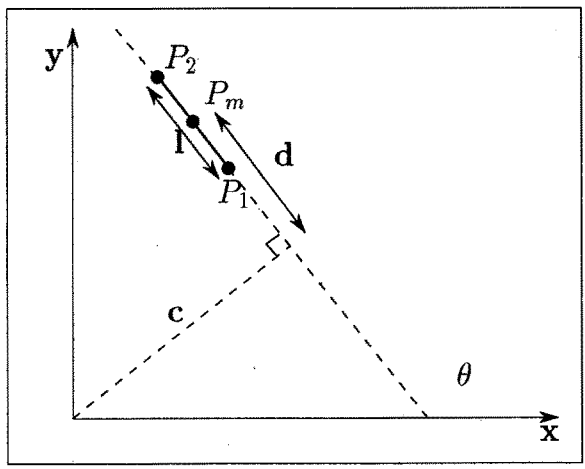

Figure 1: *Representations c,d,θ,l and xm,ym,θ,l*

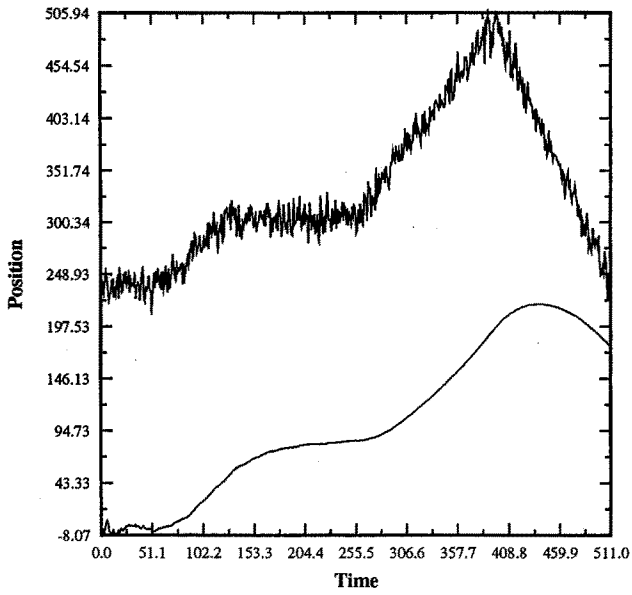

Figure 2: Noisy ramp model and its prediction by a Kalman without errors on the model

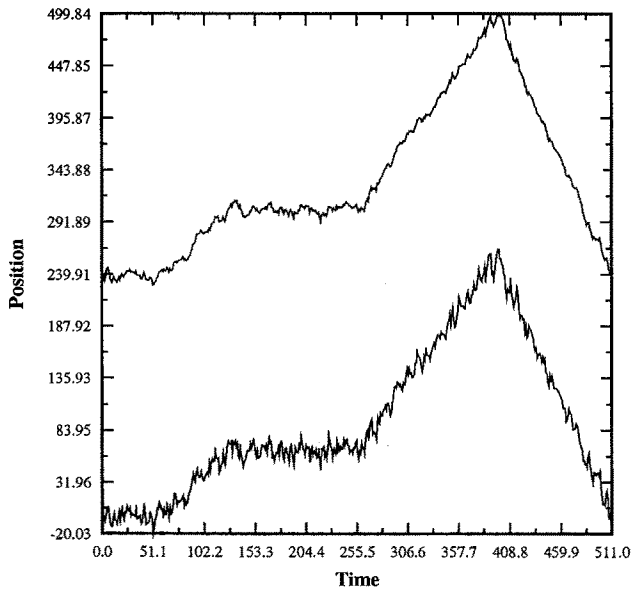

Figure 3: Prediction by a Kalman filtering with error modelling. σ_v^2*=0.001 (Up) and* σ_v^2*=0.1 (Down)*

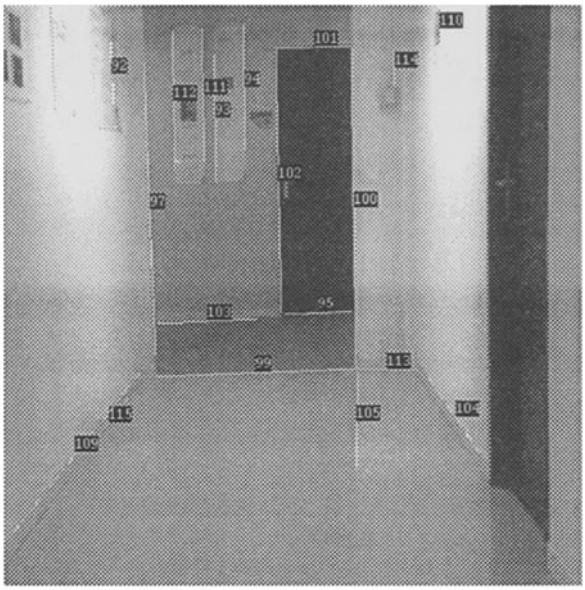

Figure 4: Hall scene 2

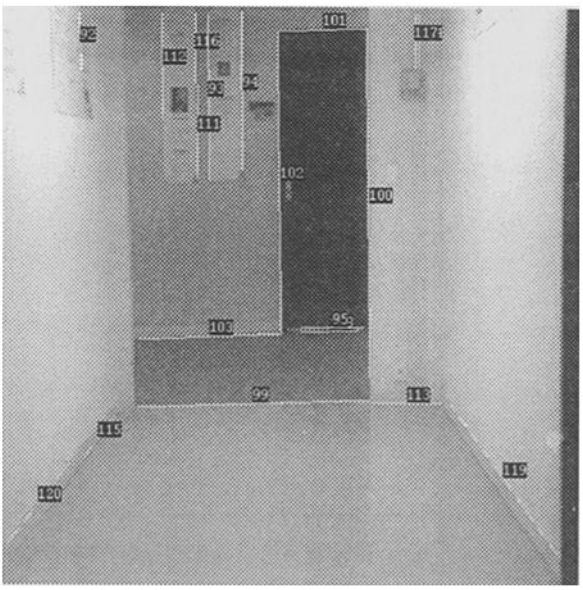

Figure 5: Hall scene 1