A Bayesian Corner Detector: Theory And Performance Evaluation

Xining Zhang, Robert M. Haralick, Visvanathan Ramesh and Anand S. Bedekar*
Intelligent Systems Laboratory
Department of Electrical Engineering, FT-10
University of Washington
Seattle, WA 98195
{xzhang,haralick,rameshv,anand}@george.ee.washington.edu

Abstract

This paper presents a corner detection method which estimates a corner to be that point on the input digital arc whose *a posteriori* probability of being a corner is the maximum among all the points on the arc. MAP estimates for the line parameters of the two intersecting lines forming the corner are also obtained. A corner is modelled as the intersection of two lines. Points on these lines are subjected to random perturbations to give rise to the observed arc segment. The perturbations on the sample points are assumed to be i.i.d Gaussian random variables of zero mean and variance σ^2. The perturbations on the points are assumed to be orthogonal to the ideal line. The paper discusses the theory of the corner detector, and extends the basic theory to handle piecewise linear arc segments by sliding a context window along the arc segment and doing a two-line-segment corner detection within each context window. Theoretical analysis for the error in the location of the estimated corner is presented. The protocol according to which experiments were conducted is described. Performance curves plotting location error versus the noise variance, the included corner angle, and the arc length, are provided. The performance of the corner detector is characterized by its false alarm rate and misdetection rates. Plots of the false alarm rate and the misdetection rate versus the included corner angle, and versus the context window length for piecewise linear arc segments, are given. Experimental results are shown to match the theoretical results. Results on real images are also presented. A protocol to characterize the performance of the corner detector on real image sets is outlined.

1 Introduction

There are two primary groups of corner detection algorithms: one is based on detection directly from the underlying image[1-5], the other one is based on detection from arcs or curves[6-19] produced from previous low level image processing operations such as edge detection or line finding followed by thinning, linking and labeling. In addition, some researchers[23] have also explored corner detection based on combinations of these methods. Corner detection on arc segments can be used to detect dominant points for curve segmentation, so that shapes of object boundaries or meaningful curves can be described either by dominant points or parameters of the segmented curves between each pair of consecutive dominant points. Further, the shape can be analyzed and recognized[6-12].

This paper presents a maximum *a posteriori* (MAP) probability corner detection method. For a given arc segment, the corner is estimated to be that point whose *a posteriori* probability of being a corner is the maximum among all the points on the arc segment. The method is not only an arc segment corner detection scheme but also provides a way of determining a polygonal approximation to a set of boundary points.

We model an ideal corner as the intersection point of two straight lines. Our mathematical formulation of the corner detection incorporates the prior distributions for corner model parameters, such as the parameters of the lines forming the corner and the index of the corner point along the arc segment. If the prior information is ignored, the method becomes a maximum likelihood corner estimator or a maximum likelihood polygonal approximation of a planar curve. Prior distributions specific to each application domain can be incorporated in our method and the corner detector can thus be tuned for specific applications.

The detection procedure involves sliding a context window of specified length over the given sequence of pixels forming the arc segment, and doing a two-line segment corner detection within each window. This context window length is chosen so that the assumption that there is only one corner within the context window holds.

The paper first discusses the theory of the two-line-segment corner detector in the Bayesian framework, and its application to multilinear segment arcs. The problem turns out to be a nonlinear optimization problem. We discuss a two step strategy for the optimization. The first step produces an initial estimate of the solution and the second step uses the gradient search to obtain the final approximation. Next, we analyze the error in the location of the estimated corner. Finally we discuss the experiments and results for the proposed approach on synthetic data as well as real data.

*Funding from DARPA contract 92-F1428000-000 is gratefully acknowledged.

2 Motivation and Theory

A corner is a discontinuity of the curvature of a curve and the location of the discontinuity can be approximated by two straight lines in its local neighborhood. The discontinuity point is called the break point or the dominant point. For a piecewise linear approximation of a curve, dominant points are also called corners. Each corner has its own local neighborhood defined by the points on the two line segments forming the corner. In general a point sequence contains multiple corners, and we would like to detect only one corner in each neighborhood specified by two intersecting line segments.

2.1 Corner Model and Its Detector

We model an ideal corner as the intersection of two straight lines. Given an observed sequence of ordered points arising from two line segments, the last observed point arising from the first estimated line segment is what we want to detect as the corner. The problem is to decide which of the points in the observed sequence has the maximum *a posteriori* probability of being the "last" point from the first line segment. The following is the formalized problem statement:

Problem Statement

Given: an observed sequence of ordered points from an arc segment, $\hat{S} = \{\begin{pmatrix} \hat{r}_i \\ \hat{c}_i \end{pmatrix} \mid i = 1, ..., I; (\hat{r}_i, \hat{c}_i) \in Z_R \times Z_C\}$, where $Z_R \times Z_C$ is the image domain, I is the number of points and $(\hat{r}_i, \hat{c}_i), i = 1, ..., I$ are the results of random perturbations on the points $(r_i, c_i), i = 1, ..., I$ constrained by

$$\begin{aligned} r_i cos\theta_1 + c_i sin\theta_1 - \rho_1 &= 0, i = 1, ..., k; \\ r_i cos\theta_2 + c_i sin\theta_2 - \rho_2 &= 0, i = k+1, ..., I, \end{aligned}$$

where $\theta_j, \rho_j; j = 1, 2$ are line orientation and location parameters for the two line segments. and k is the index of the true corner position (r_k, c_k). Assume perturbations to be independently introduced on each sample point with Gaussian distributed noise in the direction perpendicular to the line segment. Perturbations on the two line segments can be expressed by

$$\begin{aligned} \hat{r}_i &= r_i + \eta_i cos\theta_1; \hat{c}_i = c_i + \eta_i sin\theta_1; i = 1, ..., k; \\ \hat{r}_i &= r_i + \eta_i cos\theta_2; \hat{c}_i = c_i + \eta_i sin\theta_2, ; i = k+1, ..., I. \end{aligned}$$

where $\eta_i \sim N(0, \sigma^2)$.

Find: the estimated corner $(\hat{r}_{k^*}, \hat{c}_{k^*}), 2 \le k^* \le I - 1$, along the arc $\hat{S}$ and the estimates of two line parameters, (θ_1^*, ρ_1^*) and (θ_2^*, ρ_2^*) so that

$$(k^*, \theta_1^*, \rho_1^*, \theta_2^*, \rho_2^*) = \arg \max_{(k,\theta_1,\rho_1,\theta_2,\rho_2)} P(k, \theta_1, \rho_1, \theta_2, \rho_2 \mid \hat{S}, \sigma, I)$$

By Bayes' formula this can be written as

$$(k^*, \theta_1^*, \rho_1^*, \theta_2^*, \rho_2^*) = \arg \max_{(k,\theta_1,\rho_1,\theta_2,\rho_2)} P(\hat{S} \mid k, \theta_1, \rho_1, \theta_2, \rho_2, \sigma, I). \\ P(k, \theta_1, \rho_1, \theta_2, \rho_2 \mid \sigma, I). \quad (1)$$

The first term of the right side of the equation is the likelihood of observing the given sequence of points $\hat{S}$, given the parameters of the two lines forming the corner, the noise standard deviation σ, I and the index k of the true corner. The model says that the observed sequence $\hat{S}$ can be separated into two sub-sequences, or sub segments, $\hat{S}_1$ and $\hat{S}_2$, where, $\hat{S}_1 = \{\begin{pmatrix} \hat{r}_i \\ \hat{c}_i \end{pmatrix} \mid i = 1, .., k\}$ and $\hat{S}_2 = \{\begin{pmatrix} \hat{r}_i \\ \hat{c}_i \end{pmatrix} \mid i = k+1, ..., I\}$. Since perturbations on the first line (θ_1, ρ_1) are independent from those on the second line (θ_2, ρ_2), the likelihood of the observed $\hat{S}$ given two lines (θ_1, ρ_1), (θ_2, ρ_2) can be written as

$$P(\hat{S} \mid k, \theta_1, \rho_1, \theta_2, \rho_2, \sigma, I) = \\ P(\hat{S}_1 \mid k, \theta_1, \rho_1, \sigma, I) P(\hat{S}_2 \mid k, \theta_2, \rho_2, \sigma, I).$$

The perturbation model assumes that Gaussian noise is independently added onto each point of each line segment in the direction perpendicular to the line segment. The conditional probability of observing the first sub segment given the true line parameters is given by

$$\begin{aligned} & P(\hat{S}_1 \mid k, \theta_1, \rho_1, \sigma, I) \\ &= P(\hat{S}_1 \mid k, \theta_1, \rho_1, \sigma) \\ &= P((\hat{r}_1, \hat{c}_1)', ..., (\hat{r}_k, \hat{c}_k)' \mid \theta_1, \rho_1, \sigma) \\ &= \prod_{i=1}^{k} P((\hat{r}_i, \hat{c}_i)' \mid \theta_1, \rho_1, \sigma) \\ &= (\frac{1}{\sqrt{2\pi}\sigma})^k \prod_{i=1}^{k} e^{-\frac{1}{2\sigma^2}(\hat{r}_i cos\theta_1 + \hat{c}_i sin\theta_1 - \rho_1)^2}. \end{aligned}$$

Similarly, the conditional probability of observing the second sub segment $\hat{S}_2$ can be computed by

$$P(\hat{S}_2 \mid k, \theta_2, \rho_2, \sigma, I) = \\ (\frac{1}{\sqrt{2\pi}\sigma})^{I-k} \prod_{i=k+1}^{I} e^{-\frac{1}{2\sigma^2}(\hat{r}_i cos\theta_2 + \hat{c}_i sin\theta_2 - \rho_2)^2}.$$

The index k and parameters (θ_1, ρ_1), (θ_2, ρ_2) are independent of σ, hence

$$P(k, \theta_1, \rho_1, \theta_2, \rho_2 \mid \sigma, I) = P(k, \theta_1, \rho_1, \theta_2, \rho_2 \mid I).$$

Further, the index k is independent from the line parameters (θ_1, ρ_1) (θ_2, ρ_2), and these line parameters are independent of the number of points I. So

$$\begin{aligned} & P(k, \theta_1, \rho_1, \theta_2, \rho_2 \mid I) \\ &= P(\theta_2, \rho_2, \theta_1, \rho_1 \mid I) P(k \mid I) \\ &= P(\theta_2, \rho_2, \theta_1, \rho_1) P(k \mid I) \\ &= P(\theta_2 \mid \rho_2, \theta_1, \rho_1) P(\rho_2 \mid \theta_1, \rho_1) P(\rho_1 \mid \theta_1). \\ & \quad P(\theta_1) P(k \mid I) \\ &= P(\theta_2 \mid \theta_1) P(\rho_2) P(\rho_1 \mid \theta_1) P(\theta_1) P(k \mid I). \end{aligned}$$

The index k, i. e. the index of the last point arising from the first line, is assumed to be uniformly distributed between the second point and the second-from-last point, i. e.

$$P(k \mid I) = \begin{cases} K_k = 1/I-2, & 2 \leq k \leq I-1, \\ 0, & \text{otherwise.} \end{cases}$$

θ_1 is assumed to be uniformly distributed in $[0, 2\pi]$, i. e. $P(\theta_1) = 1/2\pi$.

The conditional probability distribution $P(\rho_1 \mid \theta_1)$ is a probability density of the distance ρ of the line from the origin, given θ_1, which is the orientation of the vector normal to the line.

We assume that the image domain is a square, i. e. $Z = \mid Z_R \mid = \mid Z_C \mid$ and centered at $(\mid Z_R \mid /2, \mid Z_C \mid /2)$. We also assume that the distance of the line from the origin lies in $[0 \leq \rho_1 < Z)$ (region R_I) with probability one, and in $[Z \leq \rho_1 \leq \sqrt{2}Z]$ (region R_{II}) with probability zero, and that the distance of the line to the origin has a uniform distribution in region R_I and has probability zero in region R_{II}. So the probability distribution of ρ_1 given θ_1 can be shown to be constant and equal to $1/Z$. $P(\rho_1 \mid \theta_1) = 1/Z$.

ρ_2 is assumed to be uniformly distributed in $[0 \leq \rho_2 < Z]$ and has zero probability in the region of $[Z \leq \rho_2 \leq \sqrt{2}Z]$

$$P(\rho_2) = 1/Z, 0 \leq \rho_2 < Z$$

The conditional probability distribution $P(\theta_2 \mid \theta_1)$ is assumed to be determined just by the angle included between the two lines.

$$P(\theta_2 \mid \theta_1) = P(\mid \theta_2 - \theta_1 \mid).$$

Let $\theta_{12} = \mid \theta_2 - \theta_1 \mid \in [0, \pi]$. θ_{12} is called the included corner angle. It is assumed that there is a higher probability that the included angle is close to the right angle. This assumption is consistent with some practical applications such as roof corner detection of buildings in aerial images[24-26]. We assume the probability distribution of θ_{12} to be

$$P(\theta_{12}) = K_1 e^{K_2 \sin(\theta_{12})},$$

where K_1 and K_2 are two constants.[1] K_2 can be estimated from the empirical distribution of θ_{12} by $\hat{K}_2 = 1/\hat{\sigma}^2_{\theta_{12}}$, and K_1 can be estimated by

$$\hat{K}_1 = \frac{1}{\int_0^\pi e^{\hat{K}_2 \sin(\theta_{12})} d\theta_{12}},$$

where $\hat{\sigma}^2_{\theta_{12}}$ is the estimated variance of the empirical distribution of θ_{12}.

[1]This distribution is nothing but a truncated form of the Von Mises distribution with mean$=\pi/2$.

Taking logarithms on equation 1, the problem becomes that of finding the $(k^*, \theta_1^*, \rho_1^*, \theta_2^*, \rho_2^*)$ that maximizes

$$K - \frac{1}{2\sigma^2}\sum_{i=1}^{k}(\hat{r}_i cos\theta_1 + \hat{c}_i sin\theta_1 - \rho_1)^2 - \frac{1}{2\sigma^2}\sum_{i=k+1}^{I}(\hat{r}_i cos\theta_2 + \hat{c}_i sin\theta_2 - \rho_2)^2 + $$
$$+ K_2 \sin(\mid \theta_2 - \theta_1 \mid),$$

where

$$K = \log K_1 - \log(2\pi) - 2\log Z - \log(I-2) - I\log(\sqrt{2\pi}\sigma).$$

2.2 Optimization of Parameter Estimations

The above problem is a nonlinear optimization problem. We use a two step procedure to find the solution. In the first step, we use maximum likelihood estimation to quickly find a good initial estimate and in the second step, we make use of a gradient search scheme to find the solution.

2.2.1 Initial Parameter Estimation

To maximize the posterior probability for a fixed k is to minimize

$$f(\theta_1, \rho_1, \theta_2, \rho_2) = \sum_{i=1}^{k}(\hat{r}_i \cos\theta_1 + \hat{c}_i \sin\theta_1 - \rho_1)^2 + \sum_{i=k+1}^{I}(\hat{r}_i \cos\theta_2 + \hat{c}_i \sin\theta_2 - \rho_2)^2 + g_1(\theta_1, \theta_2), \quad (2)$$

where $g_1(\theta_1, \theta_2) = -2\sigma^2 K_2 \sin(\mid \theta_2 - \theta_1 \mid)$. Setting the first derivative of f with respect to ρ_1 to zero

$$\frac{\partial f}{\partial \rho_1} = 2\sum_{i=1}^{k}(\hat{r}_i \cos\theta_1 + \hat{c}_i \sin\theta_1 - \rho_1)(-1) = 0,$$

and ρ_1 can be estimated by

$$\hat{\rho}_1 = \cos\theta_1 \bar{r}(\hat{S}_1) + \sin\theta_1 \bar{c}(\hat{S}_1), \quad (3)$$

where

$$\bar{r}(\hat{S}_1) = \frac{1}{k}\sum_{i=1}^{k} r_i; \qquad \bar{c}(\hat{S}_1) = \frac{1}{k}\sum_{i=1}^{k} c_i.$$

Similarly, ρ_2 can be estimated by

$$\hat{\rho}_2 = \cos\theta_2 \bar{r}(\hat{S}_2) + \sin\theta_2 \bar{c}(\hat{S}_2), \quad (4)$$

where

$$\bar{r}(\hat{S}_2) = \frac{1}{I-k}\sum_{i=k+1}^{I} r_i; \qquad \bar{c}(\hat{S}_2) = \frac{1}{I-k}\sum_{i=k+1}^{I} c_i.$$

Let

$$\alpha = \begin{cases} \theta_2 - \theta_1, & 0 < \theta_2 - \theta_1 \leq \pi, \\ -(\theta_2 - \theta_1), & -\pi \leq \theta_2 - \theta_1 \leq 0. \end{cases} \tag{5}$$

Therefore, the objective function can be rewritten into a function only containing two explicit parameters

$$f(\theta_1, \theta_2) = G(\theta_1, \theta_2) + \sigma^2 K_2 \sin(\alpha(\theta_1, \theta_2)),$$

where

$$G(\theta_1, \theta_2) = \tag{6}$$
$$\sum_{i=1}^{k}(\hat{r}_i \cos\theta_1 + \hat{c}_i \sin\theta_1 - \hat{\rho}_1)^2 +$$
$$\sum_{i=k+1}^{I}(\hat{r}_i \cos\theta_2 + \hat{c}_i \sin\theta_2 - \hat{\rho}_2)^2.$$

In order to decrease the domain of search for minimimizing the objective function, we can make good initial parameter guesses by obtaining θ_1 and θ_2 from the optimization only based on the two likelihood terms $G(\theta_1, \theta_2)$. Setting the first order derivative of G with respect to θ_1 to zero, we get

$$\frac{\partial f}{\partial \theta_1} = 2\sum_{i=1}^{k}(\hat{r}_i \cos\theta_1 + \hat{c}_i \sin\theta_1 - \rho_1)$$
$$(-\hat{r}_i \sin\theta_1 + \hat{c}_i \cos\theta_1) = 0, \tag{7}$$

Substituting the estimated ρ_1 from equation 3 into equation 7, we get

$$\frac{1}{2}\sin 2\theta_1[-\sum_{i=1}^{k}(\hat{r}_i - \bar{r}(\hat{S}_1))^2 + \sum_{i=1}^{k}(\hat{c}_i - \bar{c}(\hat{S}_1))^2] +$$
$$\cos 2\theta_1[\sum_{i=1}^{k}(\hat{r}_i - \bar{r}(\hat{S}_1))(\hat{c}_i - \bar{c}(\hat{S}_1))] = 0.$$

From this equation, the initial θ_1 can be estimated by

$$\hat{\theta}_1^{(0)} = \frac{1}{2}\tan^{-1}\frac{2\hat{\mu}_{rc}(\hat{S}_1)}{\hat{\mu}_{rr}(\hat{S}_1) - \hat{\mu}_{cc}(\hat{S}_1)},$$

where

$$\hat{\mu}_{rr}(\hat{S}_1) = \frac{1}{k-1}\sum_{i=1}^{k}(\hat{r}_i - (\bar{r}(\hat{S}_1)))^2,$$
$$\hat{\mu}_{cc}(\hat{S}_1) = \frac{1}{k-1}\sum_{i=1}^{k}(\hat{c}_i - (\bar{c}(\hat{S}_1)))^2,$$
$$\hat{\mu}_{rc}(\hat{S}_1) = \frac{1}{k-1}\sum_{i=1}^{k}(\hat{r}_i - \bar{r}(\hat{S}_1))(\hat{c}_i - \bar{c}(\hat{S}_1)).$$

Similarly, the initial θ_2 can be estimated by setting the derivative of G with respect to θ_2 to zero.

$$\hat{\theta}_2^{(0)} = \frac{1}{2}\tan^{-1}\frac{2\hat{\mu}_{rc}(\hat{S}_2)}{\hat{\mu}_{rr}(\hat{S}_2) - \hat{\mu}_{cc}(\hat{S}_2)},$$

where

$$\hat{\mu}_{rr}(\hat{S}_2) = \frac{1}{I-k-1}\sum_{i=k+1}^{I}(\hat{r}_i - (\bar{r}(\hat{S}_2)))^2,$$
$$\hat{\mu}_{cc}(\hat{S}_2) = \frac{1}{I-k-1}\sum_{i=k+1}^{I}(\hat{c}_i - (\bar{c}(\hat{S}_2)))^2,$$
$$\hat{\mu}_{rc}(\hat{S}_2) = \frac{1}{I-k-1}\sum_{i=k+1}^{I}(\hat{r}_i - \bar{r}(\hat{S}_2))(\hat{c}_i - \bar{c}(\hat{S}_2)).$$

Once the good initial guesses on $\hat{\theta}_1^{(0)}$ and $\hat{\theta}_2^{(0)}$ have been estimated, the location parameter $\hat{\rho}_1^{(0)}$ and $\hat{\rho}_2^{(0)}$ can be estimated by

$$\hat{\rho}_1^{(0)} = \cos\hat{\theta}_1^{(0)}\bar{r}(\hat{S}_1) + \sin\hat{\theta}_1^{(0)}\bar{c}(\hat{S}_1),$$
$$\hat{\rho}_2^{(0)} = \cos\hat{\theta}_2^{(0)}\bar{r}(\hat{S}_2) + \sin\hat{\theta}_2^{(0)}\bar{c}(\hat{S}_2).$$

2.2.2 Parameter Modifications by Gradient Search

Once the initial guesses on $\theta_1^{(0)}$, $\theta_2^{(0)}$, $\rho_1^{(0)}$ and $\rho_2^{(0)}$ have been found, then more precise estimates of these parameters can be iteratively obtained by the gradient search method. Let current estimate vector be $\hat{\theta}^{(k)} = (\hat{\theta}_1^{(k)}, \hat{\theta}_2^{(k)})'$. and

$$\nabla G(\hat{\theta}^{(k)}) = (\frac{\partial G}{\partial \theta_1}, \frac{\partial G}{\partial \theta_2})'$$

then, a new estimate vector of $\hat{\theta}^{(k+1)}$ can be approximated by

$$\hat{\theta}^{(k+1)} = \hat{\theta}^{(k)} - \lambda^{(k)} \nabla G(\hat{\theta}^{(k)}),$$

where $\lambda^{(k)}$ is a sufficiently small positive scalar and $\hat{\theta}^{(k+1)} = (\hat{\theta}_1^{(k+1)}, \hat{\theta}_2^{(k+1)})'$. Once the $\hat{\theta}^{(k+1)}$ has been updated, ρ_1 and ρ_2 can be updated by

$$\hat{\rho}_1^{(k+1)} = \cos\hat{\theta}_1^{(k+1)}\bar{r}(\hat{S}_1) + \sin\hat{\theta}_1^{(k+1)}\bar{c}(\hat{S}_1),$$
$$\hat{\rho}_2^{(k+1)} = \cos\hat{\theta}_2^{(k+1)}\bar{r}(\hat{S}_2) + \sin\hat{\theta}_2^{(k+1)}\bar{c}(\hat{S}_2).$$

In a real scenario, the iterations can be stopped by inspecting whether $\nabla G(\hat{\theta}^{(k)}) = 0$ or $\mid G(\hat{\theta}^{(k+1)}) - G(\hat{\theta}^{(k)}) \mid \leq \epsilon$, depending on which is first satisfied, where ϵ is a small constant.

When $0 < \theta_2 - \theta_1 \leq \pi$, equation (6) can be rewritten as

$$f(\theta_1, \theta_2) = G(\theta_1, \theta_2) - 2\sigma^2 K_2 \sin(\theta_2 - \theta_1).$$

In terms of equation (7) and the computations for $\hat{\mu}_{rr}(\hat{S}_j), \hat{\mu}_{cc}(\hat{S}_j)$ and $\hat{\mu}_{rc}(\hat{S}_j); j = 1, 2,,$ the partial derivatives of f are

$$\frac{\partial f}{\partial \theta_1} = A_1 \sin 2\theta_1 + B_1 \cos 2\theta_1 + 2\sigma^2 K_2 \cos(\theta_2 - \theta_1)$$
$$\frac{\partial f}{\partial \theta_2} = A_2 \sin 2\theta_2 + B_2 \cos 2\theta_2 - 2\sigma^2 K_2 \cos(\theta_2 - \theta_1)$$

where

$$\begin{aligned} A_1 &= (k-1)(\hat{\mu}_{cc}(\hat{S}_1) - \hat{\mu}_{rr}(\hat{S}_1)), \\ B_1 &= 2(k-1)\hat{\mu}_{rc}(\hat{S}_1), \\ A_2 &= (I-k-1)(\hat{\mu}_{cc}(\hat{S}_2) - \hat{\mu}_{rr}(\hat{S}_2)), \\ B_2 &= 2(I-k-1)\hat{\mu}_{rc}(\hat{S}_2), \end{aligned}$$

When $-\pi \leq \theta_2 - \theta_1 \leq 0$,

$$\frac{\partial f}{\partial \theta_1} = A_1 \sin 2\theta_1 + B_1 \cos 2\theta_1 - 2\sigma^2 K_2 \cos(\theta_2 - \theta_1),$$
$$\frac{\partial G}{\partial \theta_2} = A_2 \sin 2\theta_2 + B_2 \cos 2\theta_2 + 2\sigma^2 K_2 \cos(\theta_2 - \theta_1).$$

2.3 Application to Multi Linear Segment Models

In the above discussion, we assumed that there is only one corner in a given sequence of points. In reality, we may be provided with pixel chains that contain more than one corner. In this case, we can use the two-line-segment corner detector within a certain context window length and slide the window to perform detection on the entire arc. The procedure begins by examining the first I pixels in the chain. If a corner is detected, the corner detector puts the next context window to start at the point following the detected corner. If no corner is detected, the window is moved along the pixel chain in a fixed step size (usually one pixel) and the corner detector is reapplied.

For any context window, there is always one whose *a posteriori* probability of being a corner is the maximum among all the points in the window. However, there may actually be no true corner located in the context window. In this case, the point with the MAP probability of being a corner should not be claimed as a detected corner point. Therefore, we need a way of determining whether the MAP probability of a point being a corner is high enough so that the detector should label the point as a corner.

We first concentrate on the likelihood part of the detector. The maximization of the likelihood of observing the given sequence from the estimated corner model is equal to the minimization of

$$X_1 = \frac{1}{\sigma^2}\sum_{i=1}^{k}(\hat{r}_i \cos\theta_1 + \hat{c}_i \sin\theta_1 - \rho_1)^2 + \frac{1}{\sigma^2}\sum_{i=k+1}^{I}(\hat{r}_i \cos\theta_2 + \hat{c}_i \sin\theta_2 - \rho_2)^2 \quad (8)$$

When there is no corner in the window, $\theta_1 = \theta_2$ X_1 would be a χ^2 distributed random variable with I degrees of freedoms.

We choose a confidence level α_T and set a threshold T_p so that

$$Prob(X_1 < T_p) = \alpha_T.$$

This is the probability that a χ^2 random variable with I degrees of freedom is less than T_p. In reality, we have two estimates for the angles of the lines $\hat{\theta}_1$ and $\hat{\theta}_2$ and we compute the quantity:

$$X_2 = \min\{\frac{1}{\sigma^2}\sum_{i=1}^{I}(\hat{r}_i \cos\hat{\theta}_1 + \hat{c}_i \sin\hat{\theta}_1 - \rho_1)^2, \frac{1}{\sigma^2}\sum_{i=1}^{I}(\hat{r}_i \cos\hat{\theta}_2 + \hat{c}_i \sin\hat{\theta}_2 - \rho_2)^2\}, \quad (9)$$

and then compare X_2 with the determined threshold T_p. If X_2 is larger than T_p, the estimated break point is claimed as a detected corner. Otherwise, the estimated breaking point is not claimed as a corner.

The motivation for the criterion is as follows: if there is no corner, i. e. (θ_1, ρ_1) is same as (θ_2, ρ_2), X_1 is a χ^2 distributed random variable but X_2 is not a χ^2 distributed random variable, since the minimum of the two dependent χ^2 random variable is not χ^2 distributed. Intutively, X_2 is a statistic that computes the minimum error that could be obtained by using the left or the right line segment as the true line segment. The minimum error corresponds to the line segment $(\hat{\theta}_i, \hat{\rho}_i)$ that is the best approximation to the entire sequence. When $X_2 > T_p$, the best fit that can be obtained by fitting a single line segment to the given pixel chain gives a squared error that is more than the threshold, i. e. the probability of there being no corner is less than $1 - \alpha_T$.

If, according to the above criterion, no corner is claimed, the detector is moved along the given arc by a defined step (usually one pixel). If a corner is detected, the detector is moved to the next window starting at the pixel next to the detected corner. This procedure is repeated until the tail of the detector window reaches the last point of the given arc.

3 Location Error

The squared location error d^2 is the squared distance between the detected corner position and the true corner position,

$$d^2 = (\hat{r}_{k^*} - r^o)^2 + (\hat{c}_{k^*} - c^o)^2, \quad (10)$$

where k^* is the index of the estimated corner position (r^*, c^*), (r^o, c^o) is the true corner position, i. e. the intersection of the two lines forming the corner. Let k' be the index of the last point in the sequence that actually arises from the first line segment. From the perturbation model, when $2 \leq k^* \leq k'$,

$$\hat{r}_{k^*} = r_{k^*} + \eta_{k^*} \cos\theta_1^o; \quad \hat{c}_{k^*} = c_{k^*} + \eta_{k^*} \sin\theta_1^o. \quad (11)$$

and when $k' < k^* \leq I-2$,

$$\hat{r}_{k^*} = r_{k^*} + \eta_{k^*}\cos\theta_2^o; \quad \hat{c}_{k^*} = c_{k^*} + \eta_{k^*}\sin\theta_2^o. \quad (12)$$

where $\eta_k^* \sim N(0, \sigma^2)$.

It can be shown that

$$d^2 = (r_{k^*} - r^o)^2 + (c_{k^*} - c^o)^2 + \eta_{k^*}^2, \quad (13)$$

where $k^* \in [2, I-2]$,

Let $\Phi = (\Theta, \sigma)'$, where $\Theta = (\theta_1^o, \rho_1^o, \theta_2^o, \rho_2^o)'$. The distance d^2 is a function of the observed sequence $\hat{S}$. Its expected value is

$$E[d^2 \mid \Phi] = E[(r_{k^*} - r^o)^2 \mid \Phi] + E[(c_{k^*} - c^o)^2 \mid \Phi] + \sigma^2,$$

where the expectation is taken over all possible $\hat{S}$ that can be observed. The quantity $(r_{k^*} - r^o)^2 + (c_{k^*} - c^o)^2$ is independent of $\eta_{k^*}^2$. Therefore, the variance of the squared distance can be proved to be

$$V[d^2 \mid \Phi] = E\{[(r_{k^*} - r^o)^2 + (c_{k^*} - c^o)^2]^2 \mid \Phi\} - \{E[(r_{k^*} - r^o)^2 \mid \Phi] + E[(c_{k^*} - c^o)^2 \mid \Phi]\}^2 + 2\sigma^4,$$

where

$$E[(r_{k^*} - r^o)^2 \mid \Phi] = \sum_{i_{1^*}=2}^{I-1} (r_{k^*} - r^o)^2 P(k^*; \Phi)$$

$$E[(c_{k^*} - c^o)^2 \mid \Phi] = \sum_{i_{1^*}=2}^{I-1} (c_{k^*} - c^o)^2 P(k^*; \Phi)$$

and

$$E[\{(r_{k^*} - r^o)^2 + (c_{k^*} - c^o)^2\}^2 \mid \Phi] = \sum_{i_{1^*}=2}^{I-1} [(r_{k^*} - r^o)^2 + (c_{k^*} - c^o)^2]^2 P(k^*; \Phi)$$

where $P(k^*; \Phi)$ is the notation used to denote the probability that the observed sequence $\hat{S}$ is such that the corner estimated from it has index k^*, given that the parameters of the underlying lines are Θ and the noise standard deviation is σ.

This probability $P(k^*; \Phi)$ is estimated as follows. This is the probability that the observed sequence $\hat{S}$ is such that the corner detector detects the point with index k^* from the sequence as the corner. We approximate this probability by the probability of observing $\hat{S}$ when the true corner index, i. e. the index of the last point arising from the first line segment, is k^*. This is equal to

$$\frac{P(\hat{S}, k^* | \Phi)}{\sum_{i=2}^{I} P(\hat{S}, i | \Phi)} = \frac{P(\hat{S} | k^*, \Phi) P(k^* | \Phi)}{\sum_{i=2}^{I} P(\hat{S} | i, \Phi) P(i | \Phi)}$$

In the numerator, $P(k^*|\Phi)$ is the probability that the index of the last point arising from the first segment is k^*, given that the underlying line parameters and the noise variance are Φ. In our model, we have assumed that this index is independent of the true line parameters Θ and the noise standard deviation σ, so this probability is just the prior probability of k^* being the index of the last point arising from the first line. This prior probability was assumed uniform on integers between 2 and $I-1$.

$$P(k^*) = \begin{cases} 1/I-2, & 2 \leq k \leq I-1, \\ 0, & \text{otherwise.} \end{cases}$$

In the denominator, the terms $P(i|\Phi)$ are, by the same reasoning, all equal to $\frac{1}{I-2}$. Thus we have

$$P(k^*; \Phi) = \frac{P(\hat{S} | k^*, \Phi)}{\sum_{i=2}^{I} P(\hat{S} | i, \Phi)}$$

For each $i, 2 \leq i \leq I-1$,

$$P(\hat{S}|i, \Phi) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^I \prod_{i=1}^{j} e^{-\frac{1}{2\sigma^2}(\hat{r}_i \cos\theta_1^o + \hat{c}_i \sin\theta_1^o - \rho_1^o)^2} \cdot \prod_{j+1}^{I} e^{-\frac{1}{2\sigma^2}(\hat{r}_i \cos\theta_2^o + \hat{c}_i \sin\theta_2^o - \rho_2^o)^2}$$

The numerator $P(\hat{S}|k^*, \Phi)$ can be calculated similarly.

More rigorous analysis involves the derivation of $P(k^*|S, \Theta, \sigma, k)$, which is the probability that the corner detector detects the point having index $k*$ as the corner, given that the underlying true sequence is S, the true line parameters are Θ, σ and the index of the last point actually arising from the first line segment is k. This is the subject of another paper [?].

In a following section, experiments for examining the location error as a function of the noise standard deviation and the included corner angle are described, and experimental results are compared to theoretical results. In the experiments for examining the location error as a function of the noise standard deviation σ, for each value of σ, many trials are performed with the same value of the included corner angle. For each trial, an pair of line segments is generated and is sampled, and then the samples are perturbed with Gaussian noise of variance σ^2. For each of the N trials there is a different observed sequence $\hat{S}(n)$, $n = 1, \ldots, N$, and the observed points in $\hat{S}(n)$ are substituted in the above expression to obtain an estimate of $E(d^2|\Phi)$. Thus an estimate of $E(d^2|\Phi)$ is obtained from each trial, and these are then averaged to obtain a better estimate of $E(d^2|\Phi)$. This average is the theoretical value that is plotted in figure 1(a) as a function of

σ for constant included corner angle. In figure 1(b), the average variance of the squared location error is plotted as a function of σ.

4 Experimental Protocol and Results

This section contains three experiments: 1) the location error measurement, 2) the algorithm performance measurement, and 3) the application of the detector to real images. The first two experiments are implemented on synthetically generated sequences. The third experiment is implemented on data processed from real images.

The input parameters to the corner detector are the context window length *cwl*, the estimated standard deviation of the noise σ, and the confidence coefficient α_{T_P}.

The first two experiments utilize synthetically generated two-line-segment sequences. So we will first discuss the process of the two-line-segment sequence generation.

4.1 Two Line Segment Arc Generation

A two line segment arc can be generated in three steps:

1. Specify the starting point (r_1, c_1), the first line length L_1, the second line length L_2, the first line angle ϕ_1 and the included corner angle θ_{12}, where ϕ_1 is the counterclockwise angle between the first line and the row axis. In this step, for each line L_1 or L_2, if $|\cos\phi_j| \geq |\sin\phi_j|; j = 1, 2$, we sample the data by increasing the row coordinate by unit steps, otherwise, by increasing the column coordinate by unit steps. For the first ideal line generation, if $|\cos\phi_1| \geq |\sin\phi_1|$, then $S_1 =< (r_i, c_i) | r_i = r_1 + i;\ c_i = c_1 + i\tan(\phi_1) >; i = 0, ..., i^t$, where $i^t = \lfloor L_1 \cos\phi_1 \rfloor + 1$ otherwise, $S_1 =< (r_i, c_i) | r_i = r_1 + i\cot(\phi_1);\ c_i = c_1 + i >; i = 0, ..., i^t$, where $i^t = \lfloor L_1 \sin\phi_1 \rfloor$. For the second ideal line generation, if $|\cos\phi_2| \geq |\sin\phi_2|$, then, $S_2 =< (r_i, c_i) | r_i = L_1\cos\phi_1 + i - i^t; c_i = L_1 \sin\phi_1 + (i - i^t)\tan(\phi_2) >; i = i^t + 1, ..., I$, where $I = i^t + \lfloor L_2 \cos\phi_2 \rfloor$, otherwise, $S_2 =< (r_i, c_i) | r_i = L_1\cos\phi_1 + (i - i^t)\cot(\phi_2);\ c_i = L_1\sin\phi_1 + i - i^t >, i = i^t + 1, ..., I$, where $I = i^t + \lfloor L_2 \sin\phi_2 \rfloor$. The true corner $(r^t, c^t) = (L_1 \cos\phi_1, L_1 \sin\phi_1)$.

2. Generate a sequence of samples $< m_i; i = 1, .., I >$, where each $m_i;\ i = 1, ..., I$ is an independent random sample coming from a Gaussian distributed random variable with zero mean and a standard deviation σ.

3. Obtain a perturbed sequence of the arc segment $< (r_i + m_i\cos\theta_1, c_i + m_i \sin\theta_1); i = 1, \ldots, i^t >$, and $< (r_i + m_i \cos\theta_2, c_i + m_i \sin\theta_2); i = i^t + 1, \ldots, I >$.

Thus, a perturbed two line segment arc $\hat{S} =< (\hat{r}_i, \hat{c}_i); i = 1, ..., I >$ is generated.

4.2 Experiment One (Location Error Measurement)

In this experiment, we measure the location errors versus noise standard deviation, the included corner angle and the arc sequence length. For measuring the location error, we utilize synthetically generated two-line-segment sequences, and obtain the distance between the true corner and the detected corner. In this experiment, the context window length is equal to the length of the sequence, σ is systematically set up, and α_T is not used.

1. **Location error versus the noise standard deviation**

   - Let $\theta_{12} = 90°$, $L_1 = L_2 = 50$ units.
   - For each $\sigma \in \{0.0, 0.2, 0.4, ..., 5.0\}$ and for all of $\theta_1 \in \{0°, 1°, ..., 359°\}$, generate 10 sequences of two-line-segment arcs. There are 360*10 runs, defined as N_{run}, for each σ.
   - For each sequence, apply the corner detector, and obtain the squared location error by

   $$d_p^{2(n)} = (\hat{r}_{i_1}^{(n)} - r^{t(n)})^2 + (\hat{c}_{i_1}^{(n)} - c^{t(n)})^2,$$

   where $(\hat{r}_{i_1}^{(n)}, \hat{c}_{i_1}^{(n)}); n = 1, ..., N_{run}$ is the estimated corner and $(r^{t(n)}, c^{t(n)}); n = 1, ..., N_{run}$ is the related true corner.
   - Obtain the root-mean-square error by

   $$\bar{d_p} = (\frac{1}{N_{run}} \sum_{n=1}^{N_{run}} d_p^{2(n)})^{\frac{1}{2}},$$

   and its related variance by

   $$var(d_p) = (\frac{1}{(N_{run} - 1)} \sum_{n=1}^{N_{run}} (d_p^{2(n)} - \bar{d}_p^2)^2)^{\frac{1}{2}}.$$

   Figures ??(a) and ??(b) are respectively the root-mean-square location error and the root-mean-square variance of the location error versus the noise standard deviation. It indicates that the error linearly increases as the noise increases and the variance of the error quadratically increases as the noise increases. It indicates that the theoretical computations and the experimental results are consistent.

2. **Location error versus the included corner angle**
   This experiment is the same as above except that θ_{12} is chosen from the set $\{10°, 20°, ..., 170°\}$ and $\sigma = 1.0$

   Figure ??(c) illustrates the root-mean-square location error versus θ_{12}, and indicates that the detection has the tendency of having smaller error for 90° included corner angle and larger error for included corner angles away 90°. In addition, the rather flat region around 90° corner angle indicates that the algorithm is more stable over a large range of corner angles.

3. **Location error versus the arc sequence length**
   This experiment is the same as that in the first case but at this time the arc length is varied from 10 to 100 by a step of 10 pixels and the two line segment lengths are kept the same, ie. $L_1 = L_2$.

   Figure ??(d) is the root-mean-square location error versus the arc length. The result indicates that the algorithm is stable with different arc lengths.

4.3 Experiment two (Performance Measurement)

Once the algorithm has been designed, it's performance should be characterized [22][27]. In this experiment, we test the performance of the detector by plotting its false alarm rate and misdetection rate versus the context window length cwl, the included corner angle θ_{12} and the distance threshold d_0 which is a special parameter used during performance test.

Here, cwl is chosen smaller than the sequence length, σ is systematically set up, and $\alpha_T = 0.9$.

The false alarm rate is defined as the probability of a true noncorners being detected as a corner, ie.

$$Prob(\text{detected as corner} \mid \text{true noncorners})$$

and the misdetection rate is defined as the probability of a true corner not being detected as a corner, ie.

$$Prob(\text{not detected as corner} \mid \text{true corner}).$$

Define a circle of radius d_0, called the distance threshold, centered at a true corner. If no point exists within this circle, a misdetection happens. If the detected corner does not fall into any region centered by a true corner with the given radius d_0, this detection is claimed as a false alarm.

Define

$f_{10} = \#\{\text{misdetections}\} = \#\{\textit{true corners}$ not being detected as corners$\}$,
$f_{11} = \#\{\text{detections}\} = \#\{\textit{true corners}$ being detected as corners$\}$,
$f_{01} = \#\{\text{false alarm}\} = \#\{\textit{non-corners}$ being detected as corners$\}$,
$f_{00} = \#\{\text{non detections}\} = \#\{\textit{non-corners}$ not being detected as corners$\}$,

and define

$$\text{the false alarm rate} = \frac{f_{01}}{f_{01} + f_{00}}, \tag{14}$$

$$\text{the misdetection rate} = \frac{f_{10}}{f_{10} + f_{11}}, \tag{15}$$

1. **False alarm rate and misdetection rate versus the context window length.**

   Let $\theta_{12} = 90°$, $L_1 = L_2 = 50$ units, $T_d = 3$. For each $cwl \in \{3, 4, ..., 70\}$, where $cwl < 2 * 50$. and all of $\theta_1 \in \{0°, 1°, ..., 359°$, generate 10 sequences of two-line-segment arcs. For each context window length, there is $N_{run} = 360 * 10$ runs. For each generated curve, detect corners. Obtain the false alarm and misdetection rates.

   Figure ??(a) is the false alarm rate versus the context window length and it implies that the developed algorithm is more stable if a context window length is large enough to contain sufficient information for the estimation.

   Figure ??(b) is the misdetection rate versus the context window length and it shows that the rate drops linearly when the window length increases.

2. **False alarm rate and misdetection rate versus the included corner angle.**
   This experiment is the same as above except that the included corner angle θ_{12} at this time is varied from 1° to 179° by a step size of 1° and the context window length $cwl = 2 * 50$ unit.

   Figure ??(c) and ??(d) are the false alarm rate and the misdetection rate. They show that the algorithm has small false alarm rate and misdetection rate around the 90°.

3. **False alarm rate and misdetection rate versus the distance threshold d_0.**
   This experiment is the same above except that the distance threshold is varied from 0 to 15 pixels by a step size of 1 pixel and the included corner angle is fixed at 90°.

   Figure ??(e) and ??(f) are the false alarm rate and the misdetection rate versus the distance threshold d_0. These rates drops nonlinearly with the increase of the distance threshold d_0.

4.4 Experiment Three (Real Image Application)

In this experiment, the corner detector operates on input produced by image processing operators which produce the desired pixel chains and applied to the real image. The image datasets used were the aerial images from the RADIUS datasets. An edge detector is first run on the image, producing a set of pixel chains, on which the corner detector is run.

The input to the corner detector is produced by using a two-step edge detector. The first step involves the estimation of gradient magnitude (in the row and column directions). We use the slope facet operator, Haralick [?], for this step. The second step uses Canny's hysteresis linking procedure to perform edge linking. Canny [?], uses two thresholds:

- a high gradient threshold, T, to mark potential edge candidates and
- a low gradient threshold, T_2, that is used in order to include additional edge pixels.

The linking procedure is a boundary tracking procedure. First pixels with gradient magnitude greater than T are marked as candidate edge pixels. Then non-maxima suppression is performed by retaining only pixels whose gradients form local maxima. Starting from each potential edge candidate the procedure tracks the boundary by the examining each neighbor along the edgel (specified by the normal to the gradient direction) and including edge pixels if the gradient magnitude is greater than T_2. The tracking terminates when a candidate neighboring pixel having a gradient magnitude less than T_2 is encountered. The linking procedure produces ordered pixel chains. These chains are subsequently used as input to the corner extraction scheme. There is one parameter called *minimum chain length* Th_l to choose sequences during implementing the corner detection. If the sequence with its length less than Th_l, the sequence will not be implemented with the corner detector. In this way, we can avoid some small sequences which may be caused by perturbations or could be the sequences we are not interested.

To evaluate the performance of the corner detector on real images, we use the RADIUS model board image data set. The performance measures being examined are the false alarm rate and the misdetection rate.

To evaluate the performance of the corner detector, the data on which the corner detector is operated should conform to the assumptions made in the model. This will show the best performance the detector is capable of. For instance, the detection is performed by sliding a context window along the edge pixel chain. The detector assumes that there is no more than one corner within each context window length. If the input to the detector contains corners located much closer than the window length chosen for the detector, the corner detector will not be able to pick them up.

The protocol being followed for evaluating the performance of the corner detection on the aerial images is as follows. The groundtruth against which the output of the corner detector is compared is obtained by annotating the aerial images to delineate the edges of buildings and other structures in the image, as well as roads, shadows etc. [?]. From the annotated groundtruth data, all annotated line segments (corresponding to straight edges) are chosen. Out of these, all corners formed by line segments whose total length is greater than the context window length chosen for the experiments are selected as the set of groundtruth corners. (The length of the context window for the corner detector was chosen as 50 pixels for the initial experiments.) For each detected corner, the nearest groundtruth corner is found. If this groundtruth corner is within a distance d_0 (chosen equal to 5 pixels), the detected corner is a "hit" and the detected corner and its nearest groundtruth corner are removed from their respective lists. If there is no groundtruth corner within d_0 pixels of the detected corner, the detected corner is declared a false alarm. Every groundtruth corner such that there is no detected corner within d_0 pixels of it is declared a misdetection. In this way false alarm and misdetection rates are computed for each image.

In the initial experiments, it was observed that the false alarm rate of the corner detector was of the order of 1% or less, whereas the misdetection rate was between 40 and 70%. It was seen that a lot of the corners that were missed by the corner detector, were actually in locations where the edge detector output was inaccurate- often there was no edge detected in the place where the groundtruth indicated an edge, and sometimes improper edge linking caused a corner to be missed. Thus these figures are really performance measures of the edge detector and corner detector modules put together and not of the corner detector alone.[2]

To evaluate the performance of the corner detector alone, the input to the corner detector and the groundtruth against which the output is compared, should be such that the corner detector does not get penalised for errors made by the edge detector module. For this, the following protocol is being followed. The edge detector output is filtered such that only those pixel chains that have some "corresponding" groundtruth line segment is selected and the rest discarded. This set of pixel chains is used as the input to the corner detector. Only those groundtruth line segments that have some "corresponding" pixel chain in the edge detector output is chosen. This set of groundtruth segments is used to build the groundtruth set of corners. By filtering the input and the groundtruth in this way, we make sure that the corner detector is not made to operate on input that is corrupted because of errors made by the edge detection and linking module, and that any false alarms and misdetections are purely due to the corner detector alone.

This performance comparison is under progress and results are expected soon.

Figure ?? shows the corner detection result on real data. Figure ??(a) shows the detected edges of the 3 cut model. Figure ??(b) shows the detected corners overlaid on the original image. Figure ??(c) shows the detected edges from one of the RADIUS model board images, with the detected corners overlaid on them. In figure ??(c), some obvious corners are apparently missed by the corner detection. However, an inspection of the edge detection and linking output shows that these are in fact linking errors. The linking module fails to link together edges whose endpoints are very close, and thus what gets input to the corner detector is not one continuous edgel chain, but two disjoint arc segments. This causes the corner detector to miss the corner.

5 Conclusions

We have discussed a corner detector that is based on MAP estimation. This detector has been developed based on a corner detector in terms of two straight line segments forming the included corner angle and

[2] It is for this reason that there is a need for a robust edge detector that has significantly lower misdetection rates. Ramesh *et al* have developed one such extraction scheme [?].

the corner model can be iteratively applied to piecewise linear arcs. Not only can this method be used for corner detection on digital arcs, but the algorithm can also be applied to polygonal approximation for curves. In addition to the theory of the detector, we also provided a theoretical analysis for the error in the estimate of the corner location. Experiments showed that the theoretical and experimental results are consistent, and that this method is less sensitive to random perturbations, more robust, stable and precise. The nonlinear optimization is solved by a two step strategy : first finding good initial parameter guess and then using gradient search scheme to find the solution. More rigorous theoretical analysis of this operator has been done and future work will involve the theoretical and empirical comparison of our algorithm with traditional methods. We are in the process of evaluating the performance of our algorithm on the RADIUS model board image data set.

References

[1] Kitchen, L., and A. Rosenfeld, "Gray level corner detection", *Pattern Recognition Lett.,* **1**, pp.95-102, 1982.

[2] Zuniga, O. A., and R. M. Haralick, "Corner detection using the facet model", Proc. IEEE CVPR, pp.30-37, 1983.

[3] Guiducci, A., "Corner characterization by differential geometry techniques", *Pattern Reconition Lett.,* **8**, pp.311-318, 1988.

[4] Mehrotra, R., and S. Nichani, "Corner Detection", *Pattern Recognition,* Vol. 23, No. 11, pp.1223-1233, 1990.

[5] Deriche, R., and G. Giraudon, "Accurate corner detection: an analytical study", IEEE ICCV, 1990, pp.66-70.

[6] Pavlidis, T., "Waveform segmentation through functional approximation", *Transactions on Computers,* Vol. C-22, No. 7, 1973, pp. 689–697.

[7] —,"Algorithms for shape analysis of contours and waveforms", IEEE Trans. PAMI-2, pp.301-312, 1980.

[8] Dunham, J. G., "Optimum uniform piecewise linear approximation of planar curves", IEEE Trans. PAMI-8, pp.67-75, 1986.

[9] Ventura, J. A., and J.-M. Chen, "Segmentation of two-dimensional curve contours", *Pattern Recognition,* Vol 25. No.10, pp.1129-1140, 1992

[10] Davis, L.S., "Understanding shape: angles and sides", *IEEE Transactions on Computers,* Vol. C-26, No. 3, 1977, pp. 236–242.

[11] Freeman, H., "Shape description via the use of critical points", *Pattern Recognition,* Vol. 10, 1978, pp. 159–166.

[12] Shirai, Y., "A context sensitive line finder for recognition of polyhedra", *Artificial Intelligence,* Vol. 4, 1973, pp. 95–119.

[13] Rosenfeld, A., and E. Johnston, "Angle detection on digital curves", *IEEE Transactions on Computers,* 1973, pp. 875–878.

[14] Phillips, T.Y., and A. Rosenfeld, "An ISODATA algorithm for straight line fitting", *Pattern Recognition Letters,* Vol. 7, 1988, pp. 291–297.

[15] Haralick, R. M. and T. Y. Phillips, "Subpixel precision corner detection and localization,"

[16] Zhang, X., and R. M. Haralick, "Bayesian corner detection", *Proc. of British Machine Vision Conference* BMVC'93, University of Surrey, Guildford. U.K. 21-23 September 1993.

[17] Ramer, U., "An iterative procedure for the polygonal approximation of plane curves", *Computer Graphics and Image Processing,* Vol. 1, 1972, pp. 244–256.

[18] Leu, J.-G., and L. Chen, "Polygonal approximation of 2-D shapes through boundary merging", *Pattern Recognition Lett.,* **7** pp.231-238, 1988.

[19] Hemminger, T. L., and C. A. Pomalaza-Raez, "Polygonal representation: a maximum likelihood approach", *Comput. Vision Graphics Image Process,* **52** pp.239-247, 1990.

[20] Pei, S.-C., and C.-N., Lin, "The detection of dominant points on digital curves by scale-space filtering", Pattern Recognition Lett. Vol.25, No. 11, pp13-7-1314, 1992.

[21] Teh, C.-H., and R.T. Chin, "A scale-independent dominant point detection algorithm", in *Proceedings of CVPR,* Ann Arbor, Michigan, 1988, pp. 229–234.

[22] Haralick, R. M."Performance characterization in image analysis: Thinning, A Case in Point", Pattern Recognition Letters 13, 1992, pp. 5–12.

[23] Xie, X., R. Sudhakar, and H. Zhuang, "Corner detection by a cost minimization approach", *Pattern Recognition,* Vol. 26, No. 8. pp.1235-1243, 1993.

[24] Ramesh, V. and R.M.Haralick, "Performance characterization of Edge operators," Presented at the Special Session on Evaluation of Modern Edge Operators, Orlando SPIE Machine Vision and Robotics Conference, April 92.

[25] Haralick, R.M., *"Edge and region analysis for digital image data,"* Computer Graphics and Image Processing, Vol.12, 1980, pp. 60-73.

[26] F.J.Canny, "Finding edges and lines in images," Tech.Rep. 720, MIT AI Lab, June 1983.

[27] Ramesh, V., *et al* "Theoretical Analysis of a Bayesian Corner Finder," Document in preparation, 1994.

[28] Ramesh, V. *et al*, "Automatic Tuning Parameter Selection for Feature Extraction Sequence," *Proc. of CVPR,* Seattle, Washington, pp.672-677, June 1994.

[29] Thornton, K. *et al*, "Groundtruthing the RADIUS Model-Board Imagery," Elsewhere in these proceedings.

[30] Ramesh, V. *et al*, "A New Edge Finder based on Integrated Gradients: Theory and Performance Evaluation," Elsewhere in these proceedings.

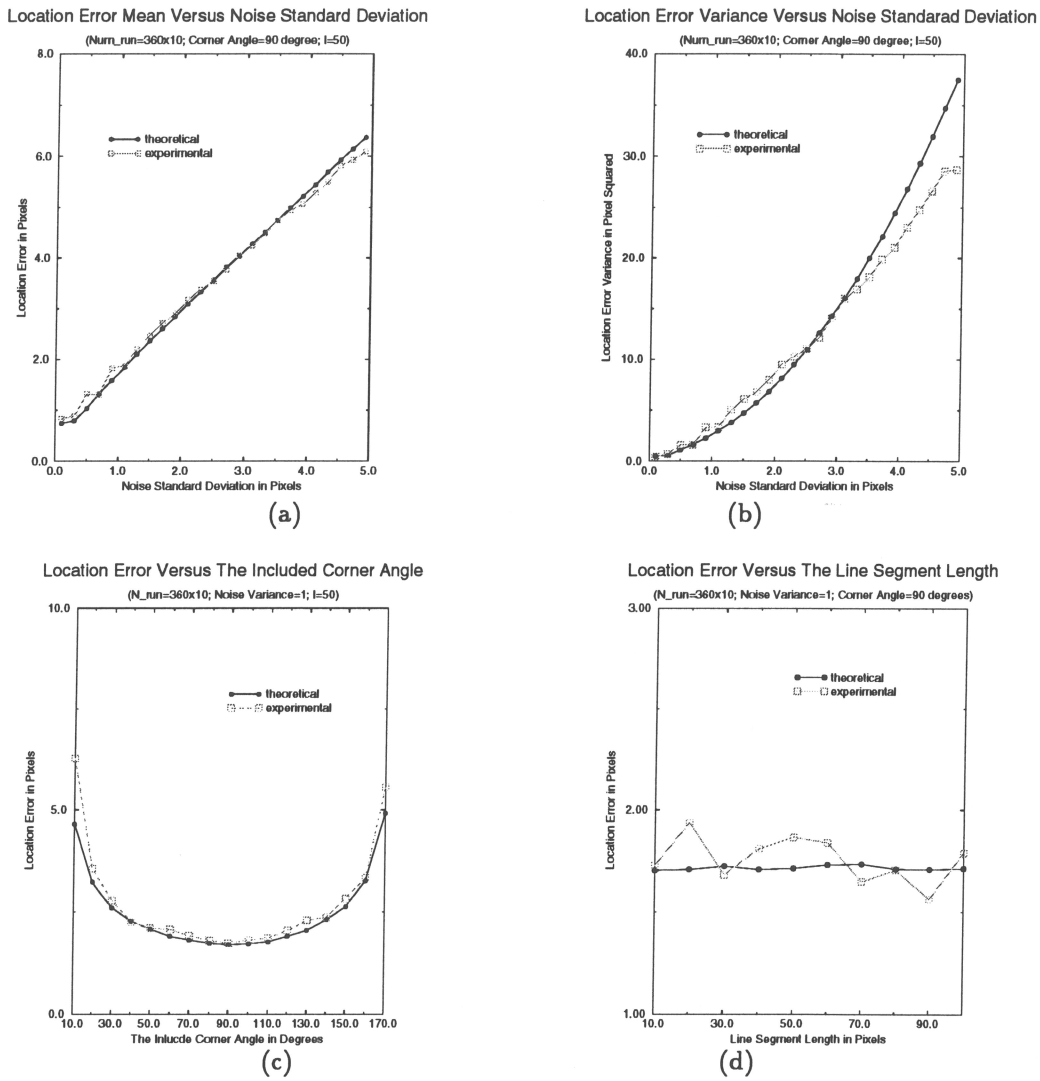


Figure 1: (a) Location error versus σ, (b) Variance of the location error versus σ, (c) Location error versus the included corner angle, and (d) Location error versus the arc length

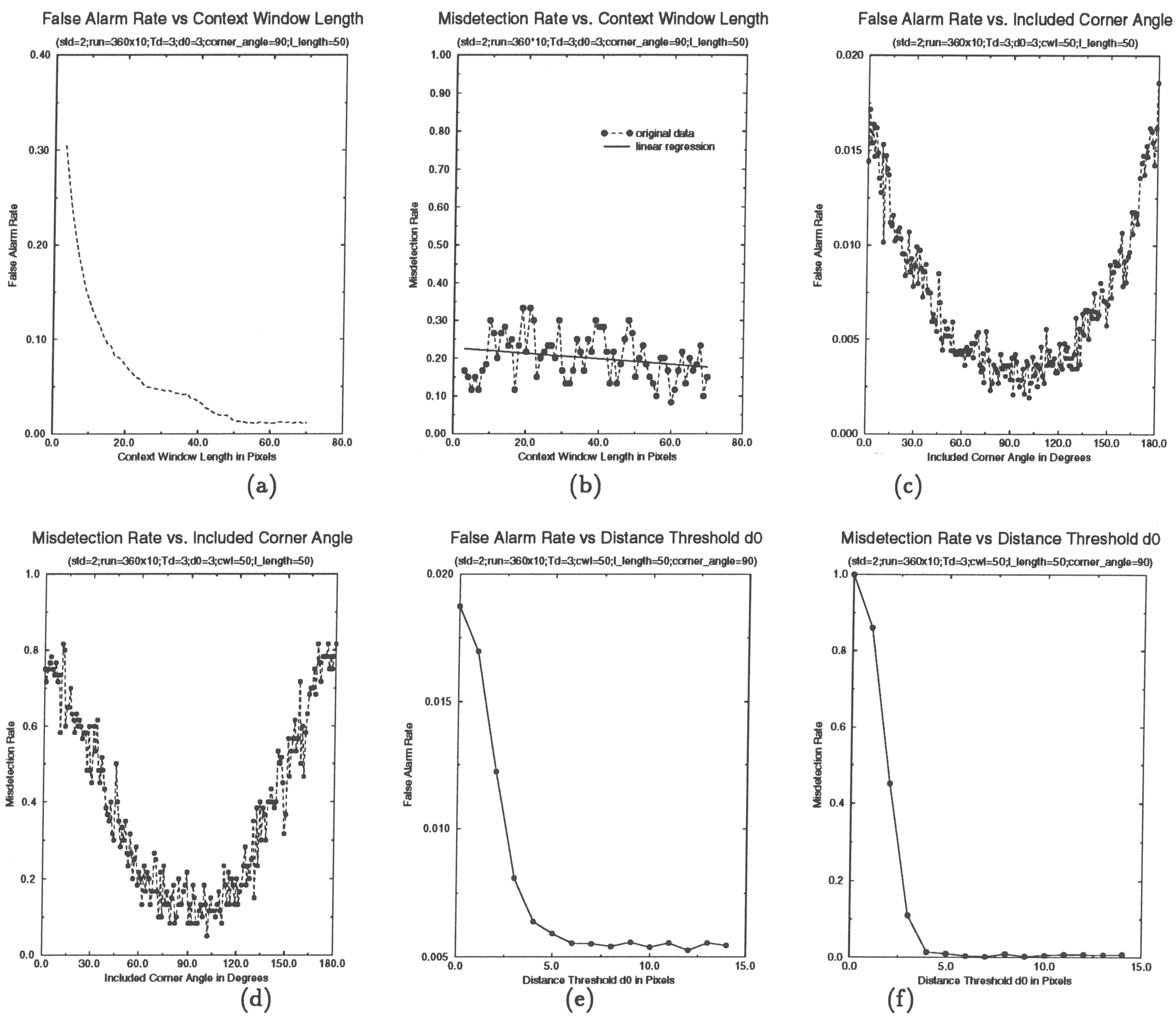
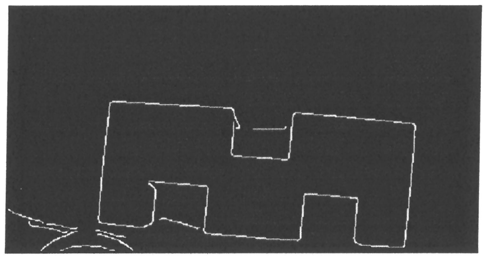
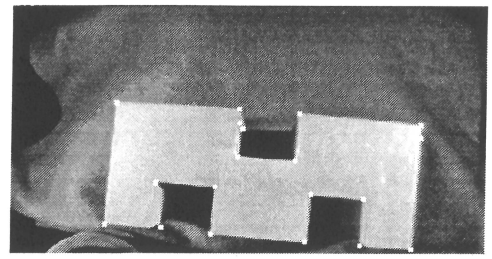


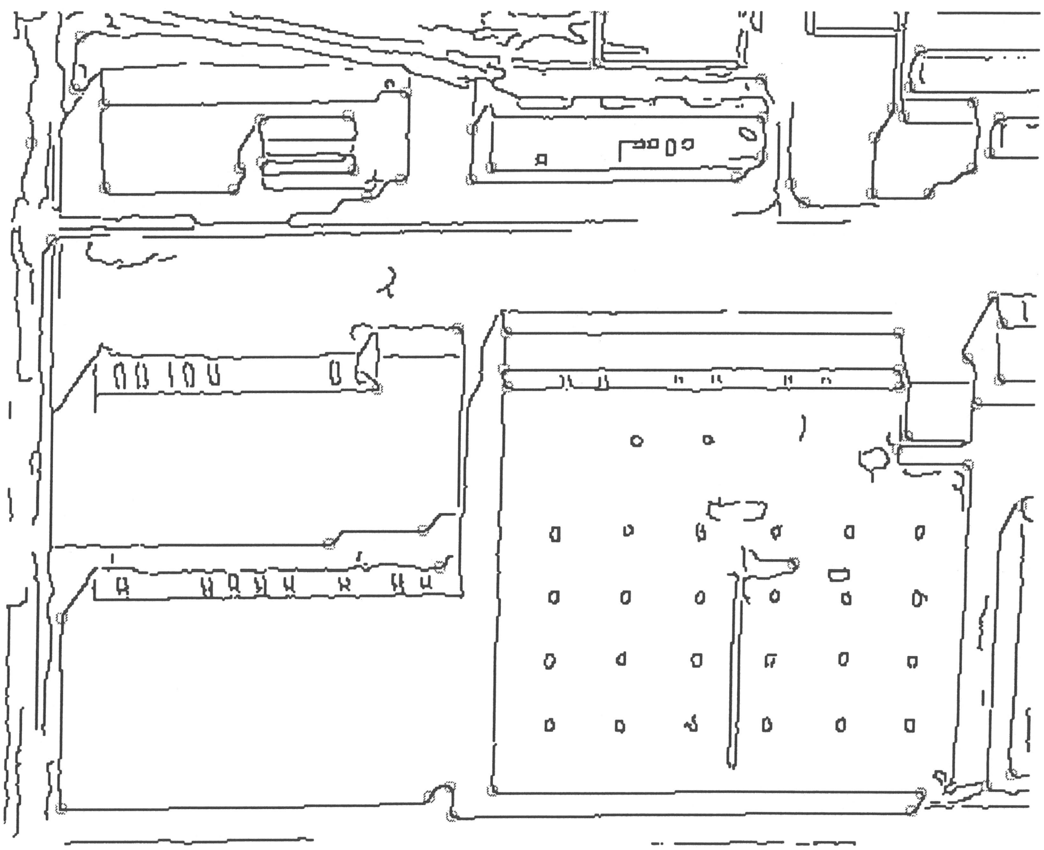
Figure 2: False alarm and Misdetection Characteristics.



(a) edges



(b) corners



(c)

Figure 3: (a) Extracted edges of the 3 cut model. (b) Detected corners overlaid on the original image. (c) Extracted edges of the building model with detected corners overlaid on them.