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  
{zhang,haralick,ramesh,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 b.i.l.d Gaussian random variables of zero mean and variance $\sigma^2$. 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[1–4], the other one is based on detection from arcs or curves[5–12] 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[22] 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[8–12].

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. The 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.
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, \( \hat{S} = \{ (x_i, y_i) \mid i = 1, \ldots, l \} \) \( (x_i, y_i) \in Z_R \times Z_G \), where \( Z_R \times Z_G \) is the image domain, \( l \) is the number of points and \( (x_i, y_i), i = 1, \ldots, l \) are the results of random perturbations on the points \( (r_i, \theta) \), \( i = 1, \ldots, l \) constrained by

\[
\begin{align*}
    r_i \cos \theta_i + c_i \sin \theta_i - r_1 &= 0, i = 1, \ldots, k, \\
    r_i \cos \theta_i + c_i \sin \theta_i - r_2 &= 0, i = k + 1, \ldots, l,
\end{align*}
\]

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

\[
\begin{align*}
    \tilde{x}_i &= x_i + \eta_i \cos \theta_i; \tilde{y}_i = y_i + \eta_i \sin \theta_i; i = 1, \ldots, k, \\
    \tilde{x}_i &= x_i + \eta_i \cos \theta_i; \tilde{y}_i = y_i + \eta_i \sin \theta_i; i = k + 1, \ldots, l,
\end{align*}
\]

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

Find: the estimated corner \((\tilde{x}_c, \tilde{y}_c)\), \(2 \leq k^* \leq l - 1\), along the arc \( \hat{S} \) and the estimates of two line parameters, \((\theta_1, \rho_1)\) and \((\theta_2, \rho_2)\) 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

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

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 \( \sigma \), \( I \) 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 \( \hat{S}_1 = \{ (x_i, y_i) \mid i = 1, \ldots, k \} \) and \( \hat{S}_2 = \{ (x_i, y_i) \mid i = k + 1, \ldots, l \} \). Since perturbations on the first line \((\theta_1, \rho_1)\) are independent from those on the second line \((\theta_2, \rho_2)\), the likelihood of the observed \( \hat{S} \) given two lines \((\theta_1, \rho_1), (\theta_2, \rho_2)\) can be written as

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

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

\[
P(\hat{S}_1 \mid k, \theta_1, \rho_1, \sigma) = P(\hat{S}_1 \mid k, \theta_1, \rho_1, \sigma) = P((x_1, y_1), \ldots, (x_k, y_k) \mid \theta_1, \rho_1, \sigma) = \prod_{i=1}^{k} P((x_i, y_i) \mid \theta_1, \rho_1, \sigma) = \left(1 - \frac{1}{2\pi\sigma^2}\right)^k \prod_{i=1}^{k} e^{-\frac{1}{2\sigma^2}(r_i \cos \theta_i + c_i \sin \theta_i - r_1)^2}.
\]

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

\[
P(\hat{S}_2 \mid k, \theta_2, \rho_2, \sigma, I) = \left(1 - \frac{1}{2\pi\sigma^2}\right)^{k-1} e^{-\frac{1}{2\sigma^2}(r_k \cos \theta_1 + c_k \sin \theta_1 - r_k)^2}.
\]

The index \( k \) and parameters \((\theta_1, \rho_1), (\theta_2, \rho_2)\) are independent of \( \sigma \), hence

\[
P(k, \theta_1, \rho_1, \theta_2, \rho_2 \mid \sigma, I) = P(k, \theta_1, \rho_1) \cdot P(\theta_2, \rho_2) \cdot P(\theta_1, \rho_1, \sigma) \cdot P(\sigma, I).
\]

Further, the index \( k \) is independent from the line parameters \((\theta_1, \rho_1), (\theta_2, \rho_2)\), and these line parameters are independent of the number of points \( I \). So

\[
P(k, \theta_1, \rho_1, \theta_2, \rho_2 \mid I) = P(k, \theta_1, \rho_1) \cdot P(\theta_2, \rho_2) \cdot P(\theta_1, \rho_1, \sigma) \cdot P(\sigma, I) = P(k, \theta_1, \rho_1) \cdot P(\theta_2, \rho_2) \cdot P(\theta_1, \rho_1, \sigma) \cdot P(\sigma, I).
\]

704
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.

\[
P(k | I) = \begin{cases} 
K_1 = 1/I - 2, & 2 \leq k \leq I - 1, \\
0, & \text{otherwise.}
\end{cases}
\]

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

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

We assume that the image domain is a square, i.e. \( Z = |Z_L| = |Z_R| \) and centered at \((|Z_L|/2, |Z_R|/2)\). We also assume that the distance of the line from the origin lies in \([0 \leq \rho_1 < Z]\) (region \( R_L \)) 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 \( R_R \) and has probability zero in region \( R_{III} \). So the probability distribution of \( \rho_1 \) given \( \theta_1 \) can be shown to be constant and equal to \( 1/Z \). \( P(\rho_1 | \theta_1) = 1/Z \).

\( \rho_2 \) 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 \( P(\theta_2 | \theta_1) \) is assumed to be determined just by the angle included between the two lines.

\[
P(\theta_2 | \theta_1) = P(|\theta_2 - \theta_1|).
\]

Let \( \theta_{12} = |\theta_2 - \theta_1| \in [0, \pi] \). \( \theta_{12} \) 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 \( \theta_{12} \) 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 \( \theta_{12} \) by \( K_2 = 1/\sigma^2_{\theta_{12}} \), and \( K_1 \) can be estimated by

\[
K_1 = \frac{1}{\int_0^\pi e^{K_2 \sin(\theta_{12})} d\theta_{12}}
\]

where \( \sigma^2_{\theta_{12}} \) is the estimated variance of the empirical distribution of \( \theta_{12} \).

\(^1\)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 \((\kappa^*, \theta_1^*, \mu_1^*, \theta_2^*, \mu_2^*)\) that maximizes

\[
K - \frac{1}{2\sigma^2} \sum_{i=1}^k (\tilde{r}_i \cos \theta_1 + \tilde{c}_i \sin \theta_1 - \rho_1)^2 - \\
\frac{1}{2\sigma^2} \sum_{i=k+1}^l (\tilde{r}_i \cos \theta_2 + \tilde{c}_i \sin \theta_2 - \rho_2)^2 + \\
+ K_2 \sin(|\theta_2 - \theta_1|),
\]

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 Estimation

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

\[
f(\theta_1, \rho_1, \theta_2, \rho_2) = \\
\sum_{i=1}^k (\tilde{r}_i \cos \theta_1 + \tilde{c}_i \sin \theta_1 - \rho_1)^2 + \\
\sum_{i=k+1}^l (\tilde{r}_i \cos \theta_2 + \tilde{c}_i \sin \theta_2 - \rho_2)^2 + g_1(\theta_1, \theta_2), (2)
\]

where \( g_1(\theta_1, \theta_2) = -2\sigma^2 K_2 \sin(|\theta_2 - \theta_1|) \). Setting the first derivative of \( f \) with respect to \( \rho_1 \) to zero

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

and \( \rho_1 \) can be estimated by

\[
\hat{\rho}_1 = \cos \theta_1 \hat{r}(\hat{S}_1) + \sin \theta_1 \hat{c}(\hat{S}_1), (3)
\]

where

\[
\hat{r}(\hat{S}_1) = \frac{1}{k} \sum_{i=1}^k r_{ij}, \quad \hat{c}(\hat{S}_1) = \frac{1}{k} \sum_{i=1}^k c_{ij}.
\]

Similarly, \( \rho_2 \) can be estimated by

\[
\hat{\rho}_2 = \cos \theta_2 \hat{r}(\hat{S}_2) + \sin \theta_2 \hat{c}(\hat{S}_2), (4)
\]

705
where
\[ \hat{\varphi}(\hat{S}_2) = \frac{1}{I-k} \sum_{i=k+1}^{I} \hat{r}_i; \quad \hat{c}(\hat{S}_2) = \frac{1}{I-k} \sum_{i=k+1}^{I} \hat{c}_i. \]

Let
\[
\alpha = \left\{ \begin{array}{ll}
\theta_2 - \theta_1, & 0 < \theta_2 - \theta_1 \leq \pi, \\
-(\theta_2 - \theta_1), & -\pi \leq \theta_2 - \theta_1 \leq 0.
\end{array} \right.
\]

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

In order to decrease the domain of search for minimizing the objective function, we can make good initial parameter guesses by obtaining \(\hat{\theta}_1\) and \(\hat{\theta}_2\) 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 \(\theta_1\) to zero, we get
\[ \frac{\partial f}{\partial \theta_1} = 2 \sum_{i=1}^{k} (\hat{r}_i \cos \theta_1 + \hat{c}_i \sin \theta_1 - \hat{\rho}_1) \left( -\hat{r}_i \sin \theta_1 + \hat{c}_i \cos \theta_1 \right) = 0, \]

Substituting the estimated \(\hat{\rho}_1\) from equation 3 into equation 7, we get
\[ \frac{1}{2} \sin 2\theta_1 \left[ - \sum_{i=1}^{k} (\hat{r}_i - \hat{\varphi}(\hat{S}_1))^2 + \sum_{i=1}^{k} (\hat{c}_i - \hat{c}(\hat{S}_1))^2 \right] + \cos 2\theta_1 \left( \sum_{i=1}^{k} (\hat{r}_i - \hat{\varphi}(\hat{S}_1)) (\hat{c}_i - \hat{c}(\hat{S}_1)) \right) = 0. \]

From this equation, the initial \(\hat{\theta}_1\) can be estimated by
\[ \hat{\theta}_1^{(0)} = \frac{1}{2} \tan^{-1} \frac{2\hat{\mu}_r(\hat{S}_1)}{\hat{\mu}_r(\hat{S}_1) - \hat{\mu}_c(\hat{S}_1)}, \]

where
\[ \hat{\mu}_r(\hat{S}_1) = \frac{1}{k-1} \sum_{i=1}^{k} (\hat{r}_i - \hat{\varphi}(\hat{S}_1))^2, \]
\[ \hat{\mu}_c(\hat{S}_1) = \frac{1}{k-1} \sum_{i=1}^{k} (\hat{c}_i - \hat{c}(\hat{S}_1))^2, \]
\[ \hat{\mu}_c(\hat{S}_1) = \frac{1}{k-1} \sum_{i=1}^{k} (\hat{r}_i - \hat{\varphi}(\hat{S}_1)) (\hat{c}_i - \hat{c}(\hat{S}_1)). \]

Similarly, the initial \(\hat{\theta}_2\) can be estimated by setting the derivative of \(G\) with respect to \(\theta_2\) to zero.
\[ \hat{\theta}_2^{(0)} = \frac{1}{2} \tan^{-1} \frac{2\hat{\mu}_c(\hat{S}_2)}{\hat{\mu}_r(\hat{S}_2) - \hat{\mu}_c(\hat{S}_2)}, \]

where
\[ \hat{\mu}_r(\hat{S}_2) = \frac{1}{I-k-1} \sum_{i=k+1}^{I} (\hat{r}_i - \hat{\varphi}(\hat{S}_2))^2, \]
\[ \hat{\mu}_c(\hat{S}_2) = \frac{1}{I-k-1} \sum_{i=k+1}^{I} (\hat{c}_i - \hat{c}(\hat{S}_2))^2, \]
\[ \hat{\mu}_c(\hat{S}_2) = \frac{1}{I-k-1} \sum_{i=k+1}^{I} (\hat{r}_i - \hat{\varphi}(\hat{S}_2)) (\hat{c}_i - \hat{c}(\hat{S}_2)). \]

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
\[ \hat{\rho}_1^{(0)} = \cos \hat{\theta}_1^{(0)} \hat{\varphi}(\hat{S}_1) + \sin \hat{\theta}_1^{(0)} \hat{c}(\hat{S}_1), \]
\[ \hat{\rho}_2^{(0)} = \cos \hat{\theta}_2^{(0)} \hat{\varphi}(\hat{S}_2) + \sin \hat{\theta}_2^{(0)} \hat{c}(\hat{S}_2). \]

2.2.2 Parameter Modifications by Gradient Search

Once the initial guesses on \(\hat{\theta}_1^{(0)}\), \(\hat{\theta}_2^{(0)}\), \(\hat{\rho}_1^{(0)}\) and \(\hat{\rho}_2^{(0)}\) 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
\[ \nabla G(\hat{\theta}^{(k)}) = \left( \frac{\partial G}{\partial \theta_1}, \frac{\partial G}{\partial \theta_2} \right), \]

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, \(\hat{\rho}_1^{(k+1)}\) and \(\hat{\rho}_2^{(k+1)}\) can be updated by
\[ \hat{\rho}_1^{(k+1)} = \cos \hat{\theta}_1^{(k+1)} \hat{\varphi}(\hat{S}_1) + \sin \hat{\theta}_1^{(k+1)} \hat{c}(\hat{S}_1), \]
\[ \hat{\rho}_2^{(k+1)} = \cos \hat{\theta}_2^{(k+1)} \hat{\varphi}(\hat{S}_2) + \sin \hat{\theta}_2^{(k+1)} \hat{c}(\hat{S}_2). \]

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

When \(0 < \theta_2 - \theta_1 \leq \pi\), equation (6) can be rewritten as
\[ f(\theta_1, \theta_2) = G(\theta_1, \theta_2) - 2\sigma^2 K_2 \sin(\theta_2 - \theta_1). \]

706
In terms of equation (7) and the computations for 
\( \mu_{\theta} (\tilde{S}_j) \) and 
\( \mu_{\alpha} (\tilde{S}_j) \); \( j = 1, 2 \), 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

\[
A_1 = (k - 1)(\mu_{\alpha} (\tilde{S}_1) - \mu_{\theta} (\tilde{S}_1)),
\]

\[
B_1 = 2(k - 1)\mu_{\alpha} (\tilde{S}_1),
\]

\[
A_2 = (I - k - 1)(\mu_{\alpha} (\tilde{S}_2) - \mu_{\theta} (\tilde{S}_2)),
\]

\[
B_2 = 2(I - k - 1)\mu_{\alpha} (\tilde{S}_2),
\]

When \( -\pi \leq \theta_2 - \theta_1 \leq 0 \),

\[
\frac{\partial f}{\partial \theta_1} = A_1 \sin 2\theta_1 + B_1 \cos 2\theta_1 - 2\sigma_2 K_2 \cos (\theta_2 - \theta_1),
\]

\[
\frac{\partial f}{\partial \theta_2} = A_2 \sin 2\theta_2 + B_2 \cos 2\theta_2 + 2\sigma_2 K_2 \cos (\theta_2 - \theta_1),
\]

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 
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 maximisation of the likelihood of observing 
the given sequence from the estimated corner model 
is equal to the minimisation of

\[
X_1 = \frac{1}{\sigma^2} \sum_{i=1}^{I} (\tilde{r}_i \cos \theta_1 + \tilde{c}_i \sin \theta_1 - \rho_1)^2 +
\frac{1}{\sigma^2} \sum_{i=k+1}^{I} (\tilde{r}_i \cos \theta_2 + \tilde{c}_i \sin \theta_2 - \rho_2)^2
\]

When there is no corner in the window, \( \theta_1 = \theta_2 = X_1 \) 
would be a \( \chi^2 \) distributed random variable with \( I \) 
degrees of freedom.

We choose a confidence level \( \alpha \) and set a threshold 
\( T_\alpha \) so that

\[
\text{Prob}(X_1 < T_\alpha) = \alpha.
\]

This is the probability that a \( \chi^2 \) random variable 
with \( I \) degrees of freedom is less than \( T_\alpha \). In reality, we have 
two estimates for the angles of the lines \( \theta_1 \) and \( \theta_2 \) and we compute the quantity:

\[
X_2 = \min \left( \frac{1}{\sigma^2} \sum_{i=1}^{I} (\tilde{r}_i \cos \theta_1 + \tilde{c}_i \sin \theta_1 - \rho_1)^2, \right.
\]

\[
\left. \frac{1}{\sigma^2} \sum_{i=k+1}^{I} (\tilde{r}_i \cos \theta_2 + \tilde{c}_i \sin \theta_2 - \rho_2)^2 \right),
\]

and then compare \( X_2 \) with the determined threshold 
\( T_\alpha \). If \( X_2 \) is larger than \( T_\alpha \), 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. \( (\theta_1, \rho_1) \) is same as \( (\theta_2, \rho_2) \), 
\( X_1 \) is a \( \chi^2 \) distributed random variable but \( X_2 \) is not a 
\( \chi^2 \) distributed random variable, since the minimum 
of the two dependent \( \chi^2 \) random variable is not \( \chi^2 \) 
distributed. Intuitively, \( X_2 \) 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 
\( (\theta_1, \rho_1) \) that is the best approximation to the entire 
sequence. When \( X_2 > T_\alpha \), 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 \( 1 - \alpha \).

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 estimated corner position and the 
true corner position,

\[
d^2 = (\hat{r}_k - r^c)^2 + (\hat{c}_k - c^c)^2,
\]

where \( k^* \) is the index of the estimated corner position 
\( (r^c, c^c) \). \( (r^c, c^c) \) 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 \( 2 \leq k^* \leq k' \),

\[
\hat{r}_{k^*} = r_{k^*} + \eta_{k^*} \cos \theta_1^c; \quad \hat{c}_{k^*} = c_{k^*} + \eta_{k^*} \sin \theta_1^c.
\]
and when \( k' < k^* \leq I - 2 \),
\[
\hat{r}_{k'} = r_{k'} + \eta_{k'} \cos \theta_{i}^2, \quad \hat{c}_{k'} = c_{k'} + \eta_{k'} \sin \theta_{i}^2.
\]
(12)

where \( \eta_{k'} \sim N(0, \sigma^2) \).

It can be shown that
\[
d^2 = (r_{k'} - r^*)^2 + (c_{k'} - c^*)^2 + \eta_{k'}^2,
\]
where \( k^* \in [2, I - 2] \).

Let \( \Phi = (\Theta, \sigma) \), where \( \Theta = (\theta_1^0, \rho_1^0, \theta_2^0, \rho_2^0)' \). The distance \( d^2 \) is a function of the observed sequence \( \hat{S} \). Its expected value is
\[
E[d^2 | \Phi] = E[(r_{k'} - r^*)^2 | \Phi] + E[(c_{k'} - c^*)^2 | \Phi] + \sigma^2,
\]
where the expectation is taken over all possible \( \hat{S} \) that can be observed. The quantity \( (r_{k'} - r^*)^2 + (c_{k'} - c^*)^2 \) is independent of \( \eta_{k'}^2 \). Therefore, the variance of the squared distance can be proved to be
\[
V[d^2 | \Phi] = E[[r_{k'} - r^*)^2 + (c_{k'} - c^*)^2 | \Phi] - \{E[(r_{k'} - r^*)^2 | \Phi] + E[(c_{k'} - c^*)^2 | \Phi] + \sigma^2,
\]
where
\[
E[(r_{k'} - r^*)^2 | \Phi] = \sum_{i=2}^{I-1} (r_{k'} - r^*)^2 P(k^*; \Phi)
\]
\[
E[(c_{k'} - c^*)^2 | \Phi] = \sum_{i=2}^{I-1} (c_{k'} - c^*)^2 P(k^*; \Phi)
\]
and
\[
E[(r_{k'} - r^*)^2 + (c_{k'} - c^*)^2 | \Phi] = \sum_{i=2}^{I-1} [(r_{k'} - r^*)^2 + (c_{k'} - c^*)^2]^2 P(k^*; \Phi)
\]

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 \( \Theta \) and the noise standard deviation is \( \sigma \).

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

\[
P(\hat{S}, k^*; \Phi)
\]
\[
\sum_{i=2}^{I} P(\hat{S}, i; \Phi)
\]

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

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 \( \Phi \). In our model, we have assumed that this index is independent of the true line parameters \( \Theta \) and the noise standard deviation \( \sigma \), 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 \( I - 1 \).

\[
P(k^*; \Phi) = \begin{cases} 
\frac{1}{I - 2}, & 2 \leq k \leq I - 1, \\
0, & \text{otherwise}.
\end{cases}
\]

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

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

For each \( i, 2 \leq i \leq I - 1 \),

\[
P(\hat{S}|i, \Phi) = \left( \frac{1}{\sqrt{2\pi}\sigma} \right)^I \prod_{i=1}^{I} e^{-\frac{1}{2\sigma^2} (f_i \cos \theta_i^0 + \delta_i \sin \theta_i^0 - \rho_i^0)^2} \prod_{j=1}^{I} e^{-\frac{1}{2\sigma^2} (f_j \cos \theta_j^0 + \delta_j \sin \theta_j^0 - \rho_j^0)^2}
\]

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 \( \Theta \), \( \sigma \) and the index of the last point actually arising from the first line segment is \( k^* \). This is the subject of another paper [26].

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 \( \sigma \), for each value of \( \sigma \), many trials are performed with the same value of the included corner angle. For each trial, an array of line segments is generated and is sampled, and then the samples are perturbed with Gaussian noise of variance \( \sigma^2 \). 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 cul, the estimated standard deviation of the noise σ, and the confidence coefficient α2

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 (r1, c1), the first line length L1, the second line length L2, the first line angle φ1 and the included corner angle θ12, where φ1 is the counterclockwise angle of the first line and the row axis. In this step, for each line L1 or L2, 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 \( S_1 = \{(r_{i1}, c_{i1}) | r_{i1} = r_1 + i; c_{i1} = c_1 + i \tan(\phi_1) \}; i = 0, ..., i^* \), where \( i^* = \lfloor L_1 \cos \phi_1 + 1 \rfloor \). For the second ideal line generation, if \( |\cos \phi_2| \geq \sin \phi_2 \), then \( S_2 = \{(r_{i2}, c_{i2}) | r_{i2} = r_2 + i \cot(\phi_2); c_{i2} = c_1 + i \}; i = 0, ..., i^* \), where \( i^* = \lfloor L_2 \cos \phi_2 \rfloor \).

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 \( (r_{i1} + m_i \cos \theta_1, c_{i1} + m_i \sin \theta_1); i = 1, ..., i^* > \), and \( (r_{i2} + m_i \cos \theta_2, c_{i2} + m_i \sin \theta_2); i = i^* + 1, ..., I \).

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 α2

1. Location error versus the noise standard deviation

- Let \( \theta_{12} = 90^\circ, L_1 = L_2 = 50 \) units.
- For each \( \sigma \in \{0.0, 0.2, 0.4, ..., 5.0\} \) and for all of \( \theta_1 \in \{0^\circ, 1^\circ, ..., 359^\circ\} \), generate 10 sequences of two-line-segment arcs. There are 380*10 runs, defined as \( N_{\text{run}} \), for each \( \sigma \).
- For each sequence, apply the corner detector, and obtain the squared location error by
\[
\bar{d}_p^2 = (\hat{r}_{i1} - r(n))^2 + (\hat{c}_{i1} - c(n))^2,
\]

where \( (\hat{r}_{i1}, \hat{c}_{i1}); n = 1, ..., N_{\text{run}} \) is the estimated corner and \( (r(n), c(n)); n = 1, ..., N_{\text{run}} \) is the related true corner.
- Obtain the root-mean-square error by
\[
\bar{d}_p = \left( \frac{1}{N_{\text{run}}} \sum_{n=1}^{N_{\text{run}}} d_p(n) \right)^{\frac{1}{2}},
\]
and its related variance by
\[
\text{var}(\bar{d}_p) = \left( \frac{1}{N_{\text{run}} - 1} \sum_{n=1}^{N_{\text{run}}} (d_p(n) - \bar{d}_p)^2 \right)^{\frac{1}{2}}.
\]

Figures 1(a) and 1(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 \( \theta_{12} \) is chosen from the set \( \{10^\circ, 20^\circ, ..., 170^\circ\} \) and \( \sigma = 1.0 \).

Figure 1(c) illustrates the root-mean-square location error versus \( \theta_{12} \), 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, i.e. $L_1 = L_2$.

Figure 1(c) 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 Measuremement)

Once the algorithm has been designed, its 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 $c_{wl}$, the included corner angle $\theta_{12}$ and the distance threshold $d_0$ which is a special parameter used during performance test.

Here, $c_{wl}$ is chosen smaller than the sequence length, $\sigma$ is systematically set up, and $\sigma \tau = 0.9$.

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

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

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

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

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

$$f_{10} = \#\{	ext{misdetections} \} = \#\{\text{true corners not being detected as corners}\},$$

$$f_{11} = \#\{\text{detecteds} \} = \#\{\text{true corners being detected as corners}\},$$

$$f_{01} = \#\{\text{false alarm} \} = \#\{\text{non-corners being detected as corners}\},$$

$$f_{00} = \#\{\text{non detections} \} = \#\{\text{non-corners not being detected as corners}\},$$

and define

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

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

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

Let $\theta_{12} = 90^\circ$, $L_1 = L_2 = 50$ units, $T_2 = 3$. For each $c_{wl} \in \{3, 4, \ldots, 70\}$, where $c_{wl} < 2 \times 50$, and all of $\theta_{1} \in \{0^\circ, 1^\circ, \ldots, 359^\circ\}$, generate 10 sequences of two-line-segment arcs. For each context window length, there is $N_{run} = 360 \times 10$ runs. For each generated curve, detect corners. Obtain the false alarm and misdetection rates.

Figure 2(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 2(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 $\theta_{12}$ at this time is varied from $1^\circ$ to $179^\circ$ by a step size of $1^\circ$ and the context window length $c_{wl} = 2 \times 50$ unit.

Figure 2(c) and 2(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 $90^\circ$.

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 $90^\circ$.

Figure 2(e) and 2(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 [24], for this step. The second step uses Canny’s hysteresis linking procedure to perform edge linking. Canny [25], 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 examining each neighbor along the edge (specified by the normal to the gradient direction) and including edge pixels if the gradient magnitude is greater than $T$. The tracking terminates when a candidate neighboring pixel having a gradient magnitude less than $T$ 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 $T_h$ to choose sequences during implementing the corner detection. If the sequence with its length less than $T_h$, the sequence will not be implemented in 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. [28]. 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 detection 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. At each step, the 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 and linker module. To understand how we do this, consider that the boundaries on an image are divided into different classes. For example, there is the class of boundary lines and the class of other object boundaries; these are delineated in the annotation. There is also the class of clutter boundaries; none of these are delineated in the annotation. For each class of boundaries, consider the problem of determining the corner misdetection rate and false alarm rate.

The total number of corners misdetected for a given boundary class is composed of three terms: (1) the number of corners that do not show up even though both of its line segments are detected by the edge detector, because the linker did not link the two line segments together, and (2) the number of corners that do not show up even though both of its line segments are detected by the edge detector, because the linker did not link the two line segments together, and (3) the number of corners that are part of an edge detected digital chain sequence corresponding to a class boundary but which are missed by the corner detector.

The corner detector misdetection rate for a given class is the fraction of corners missed that are part of an edge-detected digital chain sequence corresponding to a class boundary. The false alarm rate for a given class is the number of detected corners not corresponding to any annotated corner divided by the number of edge pixels on digital chains in the class.

We are using the following procedure to measure this. The edge detector output is filtered such that only those pixel chains that have some "corresponding" groundtruth line segment in the class are selected and the rest discarded. This set of pixel chains is used as the input to the corner detector.

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

Figure 3 shows the corner detection result on real data. Figure 3(a) shows the detected edges of the 3 cut model. Figure 3(b) shows the detected corners overlaid on the original image. Figure 3(c) shows the detected edges from one of the RADIUS model board

---

2It 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 [29].
images, with the detected corners overlaid on them. For this image, a context window length of 56 pixels was used. In figure 3(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 edge linking errors. The linking module has, in some cases, failed to link together edges whose end points are very close. Thus the input to the corner detector is not one continuous edge chain, but two disjoint arc segments. This has caused 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 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


Figure 1: (a) Location error versus $\sigma$, (b) Variance of the location error versus $\sigma$, (c) Location error versus the included corner angle, and (d) Location error versus the arc length.
Figure 2: False alarm and Misdetection Characteristics.
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.