Machine Vision and Applications (1997) 9: 229–239

Machine Vision and Applications

Random perturbation models for boundary extraction sequence

Visvanathan Ramesh, Robert M. Haralick

Department of EE, FT-10, University of Washington, Seattle WA 98195, USA

Abstract. Computer vision algorithms are composed of different sub-algorithms often applied in sequence. Determination of the performance of a total computer vision algorithm is possible if the performance of each of the sub-algorithm constituents is given. The performance characterization of an algorithm has to do with establishing the correspondence between the random variations and imperfections in the output data and the random variations and imperfections in the input data. In this paper we illustrate how random perturbation models can be set up for a vision algorithm sequence involving edge finding, edge linking, and gap filling. By starting with an appropriate noise model for the input data we derive random perturbation models for the output data at each stage of our example sequence. By utilizing the perturbation model for edge detector output derived, we illustrate how pixel noise can be successively propagated to derive an error model for the boundary extraction output. It is shown that the fragmentation of an ideal boundary can be described by an alternating renewal process and that the parameters of the renewal process are related to the probability of correct detection and grouping at the edge linking step. It is also shown that the characteristics of random segments generated due to gray-level noise are functions of the probability of false alarm of the edge detector. Theoretical results are validated through systematic experiments.

Key words: Random perturbation models – Boundary extraction sequence – Computer vision algorithms – Performance characterization

1 Introduction

Computer vision algorithms are composed of different sub-algorithms often applied in sequence. Determination of the performance of a total computer vision algorithm is possible if the performance of each of the sub-algorithm constituents is given. The problem, however, is that for most published algorithms, no performance characterization has been established in the research literature. What does performance characterization mean for an algorithm that might be used in a machine vision system? The algorithm is designed to accomplish a specific task. If the input data are perfect and have no noise or no random variation, the output produced by the algorithm also ought to be perfect. Otherwise, there is something wrong with the algorithm. Thus, measuring how well an algorithm does on perfect input data is not interesting. Performance characterization has to do with establishing the correspondence of the random variations and imperfections the algorithm produces on the output data caused by the random variations and the imperfections on the input data. This means that to do performance characterization, we must first specify a model for the ideal world in which only perfect data exist. Then we must give a random perturbation model that specifies how the imperfect perturbed data arise from the perfect data. Finally, we need a criterion function that quantitatively measures the difference between the ideal output arising from the perfect ideal input and the calculated output arising from the corresponding randomly perturbed input.

In other work [13, 14], we derived theoretical expressions for performance characteristics of edge detectors and described a methodology of automated tuning of the free parameters of an algorithm sequence. In this paper we illustrate how a random perturbation model for the edge detector output can be successively propagated through an algorithm involving linking, and gap-filling. We have seen [13] that by starting with an gaussian noise model for the gray levels in the input image, the output perturbations in the edge detector response could be specified by three parameters: probability of false alarm, probability of misdetection, and the covariance matrix of the edge location error. Assuming the error model specified, we derive random perturbation models for the output data at each stage of our example sequence. Due to the fact that there are two types of errors, misdetection and false alarm, the output data consist of true feature entities and random features that appear due to spurious responses at the feature extraction step. Hence we analyze the problem in two parts, by deriving:

– Perturbation models for perturbed true feature entities in the output.

Present address and correspondence to: V. Ramesh, Siemens Corporate Research, 755 College Road, Princeton, NJ, USA

– Perturbation models for purely random feature entities that appear in the output.

The first part is directly related to the misdetection characteristics of the sequence. The second part is related to the false-alarm characteristics of the feature extraction sequence.

These random perturbation models are useful for performing model-based theoretical comparisons of the performance of vision algorithms. Parameters of these random perturbation models can be related to measures of error such as the probability of misdetection of feature units, probability of false alarm, and the probability of incorrect grouping. The issue of how these error models can be used to automate the selection of various free parameters is taken up in another study [14].

We organize this paper into two pieces, one containing the details of the perturbation model(s) of true feature entities at the output of the feature extraction sequence and the other containing the details of the perturbation model for random entities occurring at the output of the feature extraction sequence. Following the theoretical derivations, we describe an experimental protocol for validating the theory.

2 Boundary extraction: ideal data and perturbation model

Continuous domain curves (boundaries between regions) are implicit or parametric spatial objects. In our analysis we assume that the ideal boundary, C, can be specified in the continuous domain by parametric equations, that is, the coordinates of points on the curve $(r(s), c(s))$ satisfy the equations:

$$r(s) = \sum_{m=1}^{M} \alpha_m \phi_m(s) \tag{1}$$

$$c(s) = \sum_{m=1}^{M} \beta_m \phi_m(s) \tag{2}$$

where s is the arc length, $\phi_1, \ldots \phi_M$ are the given basis functions, $\alpha_1, \ldots, \alpha_M$ and $\beta_1, \ldots, \beta_M$ are the true coefficients. This ideal boundary is sampled at a discrete set of points to produce a one-pixel-wide, connected digital arc on a rectangular grid. The sampling parameter is w. The representation for the digital arc consists of the discrete ordered point sequence $(r_i, c_i, i = 1, \ldots, L)$, where L is the number of points in the digital arc sequence.

The above description for an ideal boundary just specifies the details of the underlying point set and the underlying gray tone characteristics over neighborhoods centered on the elements of the point set are not specified. Thus, one needs to specify additional information about the gray-tone characteristics of the image at the boundary locations. This information is partially specified by the ideal edge model and perturbation models. Specification of an ideal boundary model parameters and intensity edge parameters at each point of the boundary will enable us to derive ideal model parameters for edge pixels.

Consistent with our analysis in [13], we assume that the gray-tone variation across each pixel of the boundary can be adequately modelled by a ramp edge model. This ramp edge model has three parameters: the true gradient at the pixel, the orientation, and the scale (width) of the ramp. Thus: the ideal boundary model, including gray-tone characteristics in the neighborhood of the boundary pixels, consists of: $(r_i, c_i, g_i, \theta_i, s_i)$ $i = 1, \ldots, L$, where g_i is the true gradient magnitude, θ_i is the true edgel orientation and s_i is the true edge scale at the i'th pixel. For the purpose of this paper, we assume that the edge scale across a given boundary does not change with i and hence $s_i = s$, a constant scale parameter.

It is assumed that the true gradients g_i are independent samples from a prior distribution. The nature of the prior distribution and the mathematical equations describing the probability density function vary with the application domain. This assumption makes the analysis a little simpler.

Assume that there are N boundary fragments $(b_1, \ldots, b_N)$ detected that correspond to the true boundary. Then: the j'th detected boundary piece consists of a sequence of pixels $(\hat{r}_{i,j}, \hat{c}_{i,j}, i = 1, \ldots, \hat{L}_j)$ with each pixel having estimated attributes $\hat{g}_{i,j}, \hat{\theta}_{i,j}$ and $\hat{s}_{i,j}$. [1] We developed models for perturbations on each edgel in the dissertation [13]. Characteristics of the boundary fragments, i.e. the nature of the distributions of parameters describing the output boundary, are the subject of this paper. Specifically, we develop models for describing the boundary errors given the perturbation model parameters of an edge detector: the probability of misdetection p_m, probability of false alarm p_f, the distribution of $\hat{\theta}$, and the standard deviation for the edgel position error σ_e. Note that the probability of misdetection and false alarm should be computed by using the marginal distribution of the gradient estimate for the entire population of input images. This distribution can be derived for a given application domain and the derivations are given in the paper [14]. Figure 1 illustrates the boundary fragmentation model and associated notation for the parameters of the error model.

In the first part of our analysis we assume independence between edgel attribute estimates for adjacent pixels. Under this assumption, $\hat{g}_i, \hat{\theta}_i$ and $\hat{g}_{i+1}, \hat{\theta}_{i+1}$ are independent. This independence assumption is relaxed to include dependence between adjacent pixels in [13]. The dependence is in part due to the overlapping neighborhoods used in the estimation procedure. The dependence is also due to the fact that the gray-level noise is correlated.

3 Edge linking or grouping step – analysis

A simple edge-linking procedure links adjacent edge pixels together. A more sophisticated edge linker would use edge direction estimates. Neighboring edge pixels would be linked together if their spatial relationship is consistent with their edge directions and their edge directions are similar enough. Due to misdetection of some edge pixels, an entire boundary is not actually detected at the edge detector output. Instead, after edge labelling and linking, there are short boundary segments with gaps in between them. The gaps are caused by misdetected edges. A measure of performance of

[1] In our analysis we do not study the effect of the scale. Most operators assume a particular value for the scale of the ramp edge and do not estimate this parameter.

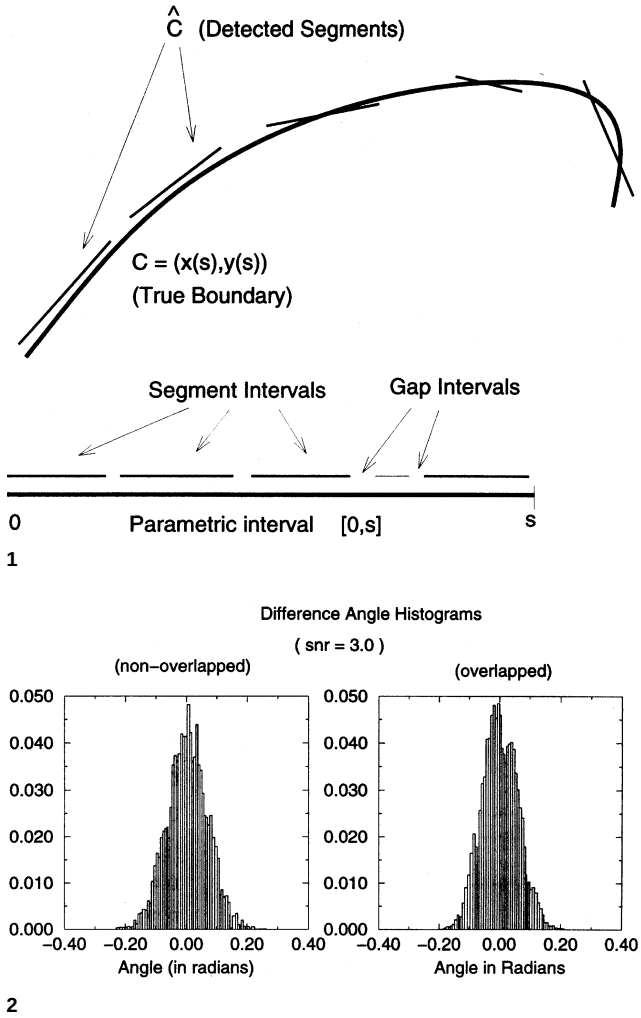


Fig. 1. The renewal process for boundary fragmentation due to noise. The segment lengths are exponentially distributed with mean parameter $1/\lambda_1$; the gap lengths are exponentially distributed with mean parameter $1/\lambda_2$. $M(s)$ denotes the expected number of breaks in the process for a given arc of length s

Fig. 2. Angle-difference histogram obtained over 1000 trials for (g/σ) = 3.0. Left histogram corresponds to the case when neighborhoods (for estimation of the angles) do not overlap (i.e., the two orientation estimates being compared are independent random variables). *Right histogram* corresponds to the case when neighborhoods overlap. Both these histograms have zero mean and the estimated precision parameter is 229.172 (theoretical $\tilde{2}25.0$) for the difference of uncorrelated angle estimates. The precision parameter for the difference of correlated angle estimates is 294.069

an edge linking scheme is the probability of correct grouping of edge pixels. We derive the expression for this probability by assuming the edge idealization specified above.

3.1 Probability of correct edge grouping

We have seen in [13] that the edgel orientation estimate is Von-Mises distributed when conditioned on the true gradient as well as the estimated gradient magnitudes and when a square neighborhood is used for the edge detector. Using this result, we can derive expressions for the probability of correct grouping of true edge pixels. Hence in the discussion that follows we assume that Θ has the Von-Mises distribution with parameters μ_0 and $\kappa = g\hat{g}/{\sigma_\alpha}^2$. Here g is the true gradient magnitude whereas $\hat{g}$ is the estimated gradient magnitude. A random variable Θ is said to be Von-Mises distributed if:

$$p(\theta) = \frac{1}{2\pi I_0(\kappa)} e^{\kappa \cos(\theta - \mu_0)} \qquad 0 \le \theta < 2\pi \qquad \kappa > 0, \quad 0 \le \mu_0 < 2\pi \tag{3}$$

Here μ_0 is the mean, κ is the precision parameter and $I_0(\kappa)$ is a modified Bessel function of the first kind, order zero and is given by:

$$I_0(\kappa) = \sum_{r=0}^{\infty} \frac{1}{r!^2} \left(\frac{1}{2}\kappa\right)^{2r} \tag{4}$$

In the case of an edge linker that links together pairs of neighboring pixels if their estimated orientations are similar enough, the difference of the estimated orientations θ_1 and θ_2 is computed. If $(\theta_1 - \theta_2) \bmod 2\pi$ is small enough, then the pixels are linked. To determine the probability of linking pairs of edge pixels whose true orientations are the same, we proceed as follows. Let θ_1 and θ_2 be Von-Mises distributed random variables with means μ_1 and μ_2 and concentration parameters κ_1 and κ_2, respectively. Then the distribution of the difference of the random variables $(\theta_1 - \theta_2) \bmod 2\pi$ is derived in Mardia [11]. It is shown in [11] that the difference is not Von-Mises distributed, but can be approximated by a Von-Mises distribution with mean $\mu_3 = \mu_1 - \mu_2$ and concentration parameter κ_3, where κ_3 is the solution to the equation:

$$A(\kappa_3) = A(\kappa_1)A(\kappa_2) \tag{5}$$

$$A(x) = 1 - \frac{1}{2x} - \frac{1}{8x^2} - \frac{1}{8x^3} + o(x^{-3}) \tag{6}$$

When $\mu_1 = \mu_2 = \mu$ and $\kappa_1 = \kappa_2 = \kappa$, then $\mu_3 = 0$ and κ_3 is the solution to the equation:

$$1 - \frac{1}{2\kappa_3} - \frac{1}{8{\kappa_3}^2} - \frac{1}{8{\kappa_3}^3} = \left\{1 - \frac{1}{2\kappa} - \frac{1}{8\kappa^2} - \frac{1}{8\kappa^3}\right\}^2 \tag{7}$$

Expanding the right-hand side and ignoring higher order terms gives the approximate solution: $\kappa_3 \approx \frac{\kappa}{2}$, which is accurate for large values of κ. The probability of the correct grouping of two pixels will therefore be given by the integral of the Von-Mises density function with parameter κ_3 over the range of allowable orientation differences. We discuss the validation of the result here in the experiment section of this paper.

4 Perturbation model at edge-detector/linker output – misdetection

Due to the misdetection of some edges, a model boundary that was supposed to be detected in its entirety appears as fragments in the edge detector/linker output. In other words an entire arc/line boundary entity was broken into several pieces with gaps between each piece. We now illustrate how the fragmentation in the boundary output can be visualized as being generated by a renewal process. In addition, we

illustrate how the various probabilities calculated at the previous section relate to the interevent distances of the renewal process.

Imagine that we start from the left of the ideal arc/line segment and walk along an infinite line. At each step the probability that the particular pixel will be labelled correctly as an edge pixel in the output is $p = 1-q$. A breakage occurs when we first encounter a pixel that is labelled incorrectly. Similarly, if we continue walking until we again encounter a pixel that is labelled correctly we would have traversed on top of a gap. If one continues walking until the end of the ideal segment is reached, one would have traversed a number of edge segments and gaps. The instances where an edge segment follows a gap can be considered as events of a discrete renewal process and the interevent times are distributed as the sum of edge segment length and the gap length.

The probability mass function for the length of the edge segment is given by:

$$P_{segmentlength}(l = k) = p^k q. \tag{8}$$

This is the geometric distribution that is a special case of the negative binomial distribution. The distribution of gap lengths between two edge segments is given by:

$$P_{gaplength}(l' = k) = q^k p. \tag{9}$$

The above distributions assume that the value for the lengths can theoretically be infinite. Let X_i be the length of the ith edge segment encountered in the walk. Let $X_i{}'$ denote the length of the ith gap along the walk.

If we assume that the true arc/line length is L pixels, we are dealing with a situation where the lengths cannot exceed L pixels. Hence the more realistic distributions would be the truncated geometric distributions. The probability mass function for the short-edge segment lengths would then be given by:

$$P_{segmentlength}(l = k) = \frac{(p^k q)}{1 - p^{L+1}}. \tag{10}$$

Similarly, the probability mass function for short gap lengths would be:

$$P_{gaplength}(l' = k) = \frac{(q^k p)}{1 - q^{L+1}}. \tag{11}$$

When an edge linker uses additional criteria than just two pixels being detected as edges and being neighbors, then the calculation of edge segment can be done, providing that the probability of linking to the previous edge pixel, given that the current pixel and the previous pixels are edge pixels, is available.

4.1 Perturbation model at edge detector/linker output – properties

In order to model the gap and segment lengths easily, we approximate the discrete distributions used in the above section by their continuous analogs. We then derive theoretical expressions for the probability density function of the interevent distances of the line-breaking process. Further, we show that the mean number of breaks in a given interval is proportional to the length of the interval, a property that is intuitively pleasing since longer segments would be more likely to be broken into pieces than shorter segments.

4.1.1 Probability density function of interval between two breaks

In this section we derive the expression for the probability density function for the interval between two breaks in the line-breaking renewal process. We know that the renewal process consists of alternating edge segments and gaps. The edge-segment lengths and the gap lengths are geometrically distributed. Here we approximate the geometric distribution by its continuous analog, the exponential distribution.

A random variable X is exponentially distributed with parameter λ if:

$$p_X(x) = \lambda exp^{-\lambda x}. \tag{12}$$

We now show that the exponential distribution is the continuous analog of the geometric distribution. The probability mass function for the geometric distribution is given by (with p being the probability of success and $q = 1 - p$.):

$$P(X = k) = p^k q \qquad k \geq 0 \tag{13}$$

Let $p = \frac{1}{1+\lambda\epsilon}$. Then the probability of having discrete lengths of t/ϵ or more is given by:

$$\sum_{k \geq \frac{t}{\epsilon}} P(X = k) = \sum_{k \geq \frac{t}{\epsilon}} \left(\frac{\lambda\epsilon}{1+\lambda\epsilon}\right)\left(\frac{1}{1+\lambda\epsilon}\right)^k$$

$$= \left(\frac{1}{1+\lambda\epsilon}\right)^{\frac{t}{\epsilon}} \tag{14}$$

If we let ϵ go to zero, we can compute the probability that a continuous random variable, X', would be greater than or equal to a specified length t. That is:

$$Prob(X' \geq t) = \lim_{\epsilon \to 0} \left(\frac{1}{1+\lambda\epsilon}\right)^{\frac{t}{\epsilon}} \tag{15}$$

$$= \exp^{-\lambda t} \tag{16}$$

This means that X' is exponentially distributed. The parameter λ of the exponential distribution is equal to q/p.

Now we model the breaks by assuming that the arc segment lengths and the gap lengths are exponentially distributed with rate parameters λ_1, and λ_2, respectively. The parameter λ_1 is an indirect measure of how often a line or curve of fixed length would break up since it is related to the mean segment length. The parameter λ_2 is a measure of how long these breaks would be.

Let X denote the random variable giving the distance between two successive starting points of short edge segments. First we derive the expression for the probability density function for the random interval X. We then derive the expression for the mean number of breaks in a line/curve of length L in the subsequent section. Let X_1 be an exponential random variable, $E(\lambda_1)$, with rate parameter λ_1. Let X_2 be an exponential random variable with rate parameter λ_2, $E(\lambda_2)$. Since $\frac{1}{\lambda_2}$ corresponds to the mean gap length,

$\lambda_2 >> \lambda_1$. We know that: $X = X_1 + X_2$. The probability density function of X is therefore the convolution of the individual probability densities of X_1 and X_2. Therefore:

$$p_X(x) = \int_0^x \lambda_1 e^{-\lambda_1 y} \lambda_2 e^{-\lambda_2 (x-y)} dy \tag{17}$$

Simplifying, this expression results in:

$$p_X(x) = \frac{\lambda_1 \lambda_2}{\lambda_2 - \lambda_1} \left(e^{-\lambda_1 x} - e^{-\lambda_2 x} \right) \tag{18}$$

The probability distribution function for X can be shown to be:

$$Prob(X \leq x) = 1 - \left(\frac{1}{\lambda_2 - \lambda_1} \right) \left(\lambda_2 e^{-\lambda_1 x} - \lambda_1 e^{-\lambda_2 x} \right) \tag{19}$$

4.1.2 Derivation for expected number of breaks in a line/curve

In the discussion that follows we derive expressions for the mean number of breaks by considering each break to be an event with the interevent distances being distributed according to Eq. 18.

Let $Y_1, Y_2, \ldots, Y_M$ be i.i.d random variables with probability density function given by Eq. 18. Let $N(t)$ denote a counting process that gives the number of breaks in an interval t. The process generated is a renewal process with probability density function for each interval being:

$$p_Y(y) = \frac{\lambda_1 \lambda_2}{\lambda_2 - \lambda_1} \left(e^{-\lambda_1 y} - e^{-\lambda_2 y} \right) \tag{20}$$

Let W_i denote the random variable giving the sum of the random event intervals until the ith event. Then: $W_i = Y_1 + Y_2 + \ldots\ldots + Y_i$. We let $h(y)$ denote the probability density function of the i.i.d. random variables Y_i. We use $H(y)$ to denote the probability distribution function of the i.i.d. random variables Y_i. We use $h^{(k)}(y)$ to denote the function obtained by k-fold convolution of $h(y)$. The Laplace transform of the probability density function of W_i, denoted as $h^{(k)}(s)$, is given by:

$$h^{(k)}(s) = \left(\frac{\lambda_1 \lambda_2}{(s + \lambda_1)(s + \lambda_2)} \right)^k \tag{21}$$

Using the inverse Laplace transform tables in [1], we get:

$$h^{(k)}(y) = \frac{(\lambda_1 \lambda_2)^k}{\Gamma(k)} \sqrt{\pi} \left(\frac{y}{\lambda_2 - \lambda_1} \right)^{k - \frac{1}{2}} e^{-(\lambda_1 + \lambda_2) y/2} \times I_{k-\frac{1}{2}} \left(\frac{\lambda_2 - \lambda_1}{2} y \right) \tag{22}$$

Here $\Gamma(k)$ is the Gamma function and $I_{k-\frac{1}{2}}(x)$ is the modified spherical Bessel function of the first kind.

The expected number of events in an interval of t is given by:

$$M(t) = \overline{N(t)} = \sum_{k=0}^{\infty} k Prob(N(t) = k) \tag{23}$$

Using $H^{(k)}(y)$ to denote the distribution function corresponding to the density function $h^{(k)}(y)$, the probability that exactly k events (or breaks) occurred in an interval of t is given by the difference $H^{(k)}(t) - H^{(k+1)}(t)$. Therefore:

$$M(t) = \sum_{k=1}^{\infty} k \left(H^{(k)}(t) - H^{(k+1)}(t) \right) = \sum_{k=1}^{\infty} H^{(k)}(t) \tag{24}$$

Taking the Laplace tranforms on both sides of the above expression, we can find that:

$$M(s) = \frac{h(s)}{s(1 - h(s))} \tag{25}$$

where $h(s)$ is the Laplace transform of the density function $h(y)$. It follows that:

$$M(s) = \frac{\lambda_1 \lambda_2}{s^2 (s + \lambda_1 + \lambda_2)} \tag{26}$$

Taking the inverse Laplace transforms on both sides of the above equation, we get:

$$M(t) = \left(\frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} \right) \left[t - \frac{1 - e^{-(\lambda_1 + \lambda_2) t}}{\lambda_1 + \lambda_2} \right] \tag{27}$$

From the above expression we can see that if t is zero $M(t)$ is zero and as t tends towards infinity $M(t)$ also approaches infinity. Normally, λ_1 is very small compared to λ_2 and under this circumstance the expression for $M(t)$ becomes:

$$M(t) = \lambda_1 \left[t - \frac{(1 - e^{-\lambda_2 t})}{\lambda_2} \right] \tag{28}$$

If the second exponential term is small compared to 1 (which will be the case since λ_2 is large for small noise levels), then:

$$M(t) = \lambda_1 t. \tag{29}$$

This means that the mean number of breaks in the line is equal to the ratio of the length of the entire line to the mean value of line-segment length.

5 Gap filling algorithm – analysis

After edge linking, the boundary gaps must be filled. The boundary gap-filling procedure will fill gaps of length less than a specified length L. The perturbation model for the input data is nothing but the renewal process, and the questions are: What is the data model for the output of the gap filling algorithm? What is the distribution of gap lengths and what is the distribution of segment lengths? We show here that the mean number of gaps that are left unfilled in the output is the product of the mean number of gaps in the input and the probability that a random gap is not filled.

Edge or gap filling can be thought of as a process that fills in gaps of lengths less than some threshold, say L. For the analysis, we assume that we have a broken-line segment with the gap length and line length parameters of λ_2 and λ_1, respectively. Assume that the line-breaking process is a renewal process with event interval length density function

as given in Eq. 18. The problem is to find the output distribution lengths obtained by filling gaps in the input of length less than L given that the length of the ideal input segment is t. Let q denote the probability that the gap length in the input is less than L. Then:

$$q = \int_0^L \lambda_2 e^{-\lambda_2 y} dy \qquad = 1 - e^{-\lambda_2 L} \tag{30}$$

If there are n gaps in the input then the probability that exactly i gaps will be filled is given by:

$$\binom{n}{i} q^i (1-q)^{n-i} \tag{31}$$

and the mean number of gaps filled will be:

$$\mu = \sum_{i=0}^{n} i \binom{n}{i} q^i (1-q)^{n-i} \qquad = nq$$
$$= n(1 - e^{-\lambda_2 L})$$

The mean number of gaps that are not filled is then given by: $ne^{-\lambda_2 L}$.

We have already seen that the mean number of breaks in a line as given by Eq. 27 is dependent on the mean line-segment lengths and mean gap-segment lengths. Here we have shown that given that there are n breaks in the input the mean number of gaps that are filled just depends on the mean gap-segment length.

The input to the gap-filling procedure is a renewal process, and the output obtained from the procedure is also a renewal process. We now derive the expression for the probability density function of the interval times in the output renewal process. Consider an interval in the output process. This interval was obtained by deleting multiple events (filling gaps of length less than some threshold) from the input process. The probability of a gap in the input being filled was $(1-e^{(-\lambda_2 L)})$. Let $p = e^{-\lambda_2 L}$. Given that there are exactly i intervals in the input process the probability that exactly $i-1$ intervals vanish to produce the output is given by: $p(1-p)^{i-1}$.

Since the output process intervals are obtained by random convolution of the interval times in the input process, the Laplace transform of the probability density function, $(h_f(s))$, for the interval time in the output process is related to the Laplace transform of the probability density function for the input process. $h_f(s)$ is related to $h(s)$ as follows:

$$h_f(s) = \sum_{i=1}^{\infty} p(1-p)^{i-1} h^i(s)$$
$$= \frac{ph(s)}{1-(1-p)h(s)}$$
$$= \frac{\lambda_1 \lambda_2 p}{s^2 + (\lambda_1 + \lambda_2)s + \lambda_1 \lambda_2 p}$$

The mean number of breaks in the output process can therefore be obtained by using the expression for $h_f(s)$. The Laplace transform of the expression for mean number of breaks can be shown to be equal to:

$$M_f(s) = \frac{\lambda_1 \lambda_2 p}{s^2(s + \lambda_1 + \lambda_2)} \tag{32}$$

Taking inverse Laplace transforms we can show that the expression for the mean number of breaks in the output of the gap filling procedure is given by:

$$M_f(t) = \left(\frac{\lambda_1 \lambda_2 p}{(\lambda_1 + \lambda_2)^2}\right)$$
$$\times \left[(\lambda_1 + \lambda_2)t - (1 - e^{-(\lambda_1+\lambda_2)t})\right] \tag{33}$$

Since $\lambda_2 >> \lambda_1$, we can approximate the above expression by setting $\lambda_1 + \lambda_2 \simeq \lambda_2$ to:

$$M_f(t) = \left(\frac{\lambda_1 p}{\lambda_2}\right)\left[\lambda_2 t - (1 - e^{-\lambda_2 t})\right]$$
$$= \lambda_1 pt - \left(\frac{\lambda_1 p}{\lambda_2}\right)\left(1 - e^{-\lambda_2 t}\right)$$

From the above expression it can be seen that the mean number of the gaps in the output is related to the mean number of gaps in the input ($\lambda_1 t$) and to the probability (p) of not filling a gap. It can also be seen that:

$$M_f(t) = M(t)p = M(t)e^{-\lambda_2 L}. \tag{34}$$

This means that the mean number of gaps in the output is the product of the expected number of gaps in the input and the probability that a gap is not filled.

5.0.3 Distribution of the gap lengths after the gap-filling process

The distribution of gap lengths after the gap-filling process is easy to derive. The gaps in the output arise only if the individual lengths are greater than the gap threshold, L. Also, the gap distribution at the input to the gap-filling algorithm was exponential with parameter λ_2. Hence: the pdf of the gap lengths at the output is given by:

$$p_X(x) = \lambda_2 e^{-\lambda_2(x-L)} \text{ if } x > L \tag{35}$$
$$= 0 \text{ otherwise}$$

5.0.4 Distribution of the edge-segment lengths after gap filling

Suppose that exactly i gaps were filled to produce a single segment in the output. Hence there are exactly $i+1$ edge segments and i gap lengths between these segments. The gap lengths were all less than L; otherwise they would not have been filled. Let $X_j, j = 1, \ldots, i+1$ denote the sequence of random variables for the edge-segment lengths in the input and $X'_k, k = 1, \ldots, i$ denote the sequence of gap lengths in the input. Then the length of a single output segment is given by:

$$\sum_{j=1}^{i+1} X_j + \sum_{k=1}^{i} X'_k. \tag{36}$$

But we know that X_j's are i.i.d exponential random variables with parameter λ_1 and X_k's are i.i.d truncated exponential random variables with probability density function:

$$p_{X_k'}(x) = \frac{\lambda_2 e^{-\lambda_2 x}}{1 - e^{-\lambda_2 L}} \text{ if } x < L \quad (37)$$
$$= 0 \text{ otherwise.}$$

It can be shown that the probability density function for the sum of $i + 1$ exponentially distributed random variables (X_j's), with parameter λ_1, is given by:

$$p_Y(y) = \frac{(\lambda_1 y)^i \lambda_1 e^{-\lambda_1 y}}{i!}$$
$$0 \le y < \infty \quad (38)$$

Similarly, it can also be shown that the pdf for the sum of i truncated exponential random variables, with parameter λ_2, is given by:

$$p_{Y'}(y) = \frac{(\lambda_2 y)^{i-1} \lambda_2 e^{-\lambda_2 y}}{(i-1)!(1 - e^{-\lambda_2 L})^i}$$
$$0 \le y \le (k-1)L \quad (39)$$

Hence the length, Z, of the output segment (given that i gaps were filled) is distributed as the sum of Y and Y'. The pdf for the sum is given by:

$$p_Z(z|i) = C_i e^{-\lambda_2 z} \times \left[\sum_{k=0}^{i} \binom{i}{k} (-1)^k z^i \int_0^z y^{i+k-1} e^{-(\lambda_2 - \lambda_1)y} dy \right]$$
$$0 \le z < \infty$$
$$C_i = \frac{{\lambda_1}^{i+1} {\lambda_2}^i}{i!(i-1)!(1 - e^{-\lambda_2 L})^i} \quad (40)$$

From this we can compute that the probability that the output segment length is z by $p_Z(z) = \sum_{i>0} p_Z(z|\text{i gaps are filled}) P(i)$, where $P(i)$ is the probability that there will be exactly i gaps filled.

6 Perturbation models for random entities

In the above analysis we modelled the perturbations to the features we are looking for, namely, curve/line segments. Everything that was said was conditioned upon the fact that a true model line/curve segment got broken into multiple pieces. Here we now focus on modelling noise due to which the algorithm sequence produces false features. For example, a sequence of correlated noise pixels gives rise to a false edge and hence a false-line segment. Typically, these line segments are of small length. A cleaning step may remove these segments, but during the process one may end up missing a number of model features.

If we had correlated noise in the gray-level image, the edge detector would label non-edge pixels as edge pixels and the edge linking and fitting step would produce short-line segments at the output. The input perturbations (due to spurious edges) are specified by the probability of false alarm of the edge detector. The set of points at the input of the edge linker can be modelled as a random point process.

Thus, in the analysis that follows we take a look at the spatial pattern of these random false labellings over an image. Since each non-edge pixel in the image gets labelled incorrectly as a true edge with probability p_f, the spatial pattern generated follows a discrete random process. The discrete random process we use is the *Bernoulli lattice process*, (see [16]). The Bernoulli lattice process is the discretized analogue of the Poisson point process.

Define w to be the width and height of each pixel and let L_w denote the lattice formed with parameter w, the mesh of the lattice. Then the Bernoulli lattice process $\phi_{p,w}$ is a random subset of L_w. Each point of L_w is contained in $\phi_{p,w}$ with probability p_f independently of all others.

If nr and nc denote the number of rows in the image and the number of columns in the image, respectively, then the mean number of pixels in the process is given by: $p_f * nr * nc$. The intensity of the process (λ) , defined as the number of points per unit area, is given by: p/w^2. If both p and w tend to zero together such that the intensity tends to λ in the limit, then the process becomes a Poisson point process.

6.0.5 Nearest-neighbor distance distribution of the point process

We have seen that the spatial process seen at the output of the edge detector is the discretized version of the Poisson point process. Since two points in the input of the edge linker are linked in the output if the distance between the points is less than a specified threshold, the distance distribution between events in the input is of interest to us. We give expressions for the nearest-neighbor distance distribution for the events of a Poisson point process here. The derivation we give is actually a special case of the general solution given in [16].

Let $D(r)$ denote the probability distribution function of the nearest-neighbor distance. Given that an event, E_1, occurred at a particular location, we wish to compute the probability that the distance to another event will be less than or equal to r. This probability is clearly equal to 1 minus the probability of finding no event within a circle of radius r. That is:

$$D(r) = 1 - P(\text{ No event within a circle of radius } r | E_1). \quad (41)$$

Since E_1 is an event of probability zero, we have to derive $D(r)$ by assuming that E_1 occurred within a circle of radius ϵ and then let ϵ tend to zero. Doing this, we can show that:

$$D(r) = 1 - e^{-(\lambda \pi r^2)}. \quad (42)$$

The mean distance can be shown to be:

$$\mu_r = \frac{1}{2\sqrt{\lambda}}. \quad (43)$$

A given edge pixel in the input has a probability $p(L)$ of getting deleted in the output, where $p(L)$ is the probability that no edge pixel exists within a radius of length L around the given edge pixel. This suggests that the input Poisson process is being thinned to produce the output process. It is shown in [16] that the process produced by dependent thinning results in a cluster-point process. However, the gap-filling algorithm fills the gaps and therefore produces segments instead of points. The process so obtained is a line-segment process.

7 Validation of theoretical results for boundary error model

This section describes the experimental protocol employed to validate the theoretical results provided in this paper. We have seen that the probability of correct edgel grouping based on similarity orientation estimates can be computed from the distribution of orientation estimate differences. This distribution was shown to be approximated by a Von-Mises distribution. We have also seen that the output error model for the boundary extraction scheme can be described by the probability distribution of segment lengths and by the probability distribution of the lengths of random segments. The errors in the pixel positions are not considered in this paper, but are also studied in [13]. Given a noisy image containing a true line feature of a particular length, we therefore obtain plots for the empirical distributions of the following measures for a given signal-to-noise ratio:

- Distribution of angle-estimate differences
- Segment-length distribution
- Gap-length distribution
- Distribution of number of breaks in true line

There are two parts to the experiment. Since the first part of our theoretical derivations is in the discrete (pixel) domain and the latter part of the derivations extends these into the continuous domain, our experiments were done both in the discrete and continuous domain. For our continuous-domain experiments we assume exponential models for the segment and gap-length distributions, simulate the alternating renewal process and validate the analytical results. For our discrete-domain experiment involving the validation of our analytical result on the distribution of angle differences, we generate synthetic-image data using the protocol described below and obtain the desired distributions. To keep the length of the paper short, the segment length and gap-length distributions in the discrete setting are not discussed and we concentrate on the results from the continuous domain experiments.

7.1 Image generation

Synthetic images of size 51 rows by 51 columns were generated with step edges at various orientations passing through the center pixel $(R, C) = (26, 26)$ in the image. The gray value, $I(r, c)$, at a particular pixel, (r, c), in the synthetic image was obtained by using the function where $\rho = (r - R)cos(\theta) + (c - C)sin(\theta)$.

$$I(r,c) = I_{min}, \quad \rho < 0 \qquad (44)$$
$$= I_{max},$$

otherwise, I_{min} and I_{max} are the gray values in the left and right of the step edge. The variables R and C designate a point in the image on which the step edge boundary lies. In our experiments we set I_{min} to be 100 and I_{max} to be 200. We used orientation (θ) values of 0, 15, ..., 175 deg. To generate ramp edges, we averaged images containing the step edges with a kernel of size 5 by 5. To these ramp edge images we added additive Gaussian noise to obtain images with various signal-to-noise ratios. We define signal-to-noise ratio as: $SNR = \frac{g}{\sigma}$. where g is the true gradient magnitude for the ramp edge and σ is the noise standard deviation. Ground-truth edge images were generated by using the following function where $\rho = (r - R)cos(\theta) + (c - C)sin(\theta)$.

$$I_1(r,c) = 0 \quad \rho < -0.5 \qquad (45)$$
$$= 1 \quad \text{otherwise.}$$
$$I_2(r,c) = 0 \quad \rho < 0.5$$
$$= 1 \quad \text{otherwise.}$$
$$I(r,c) = I_1(r,c) \text{ exor } I_2(r,c)$$

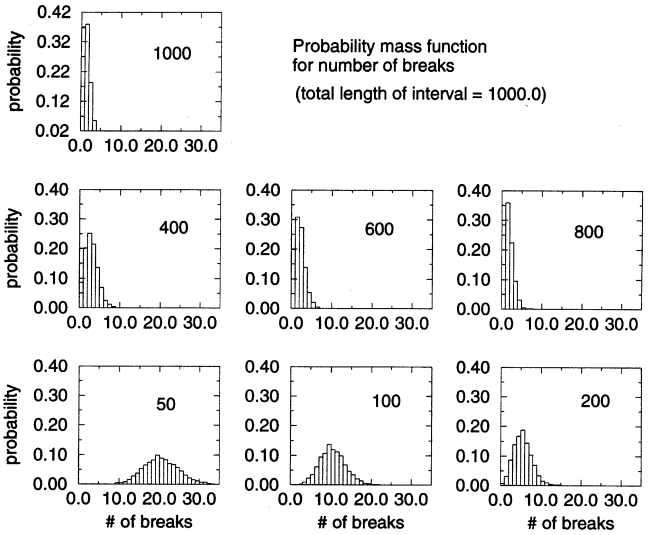
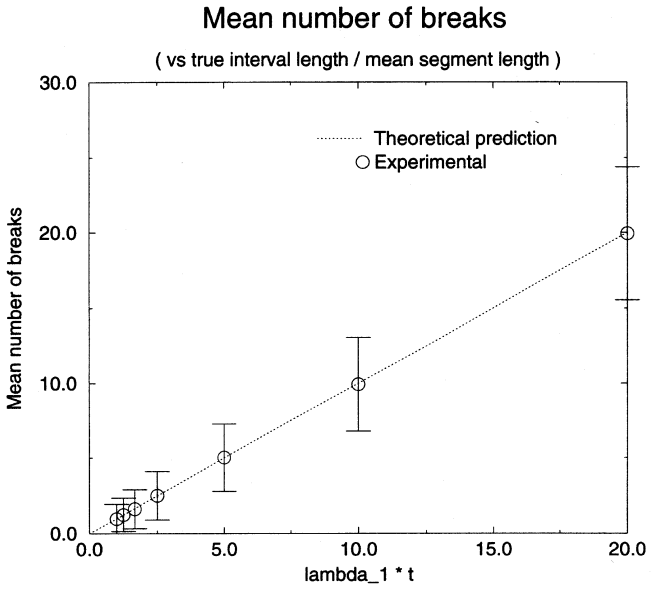


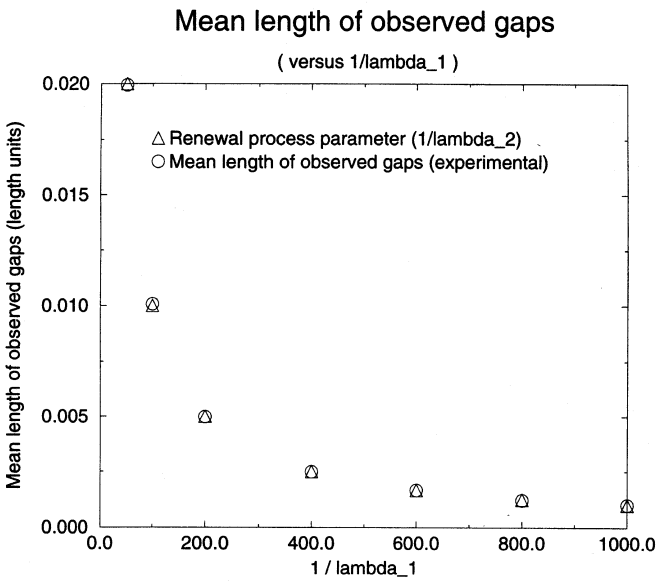
Fig. 3. Probability mass function of number of breaks in an interval of length 1000.0 units. The *number inside each graph* indicates the value of $\lambda_2 = 1/\lambda_1$ used to generate the distribution. Note how the number of breaks increases as the mean segment interval length decreases

7.2 Validation of angle difference distribution

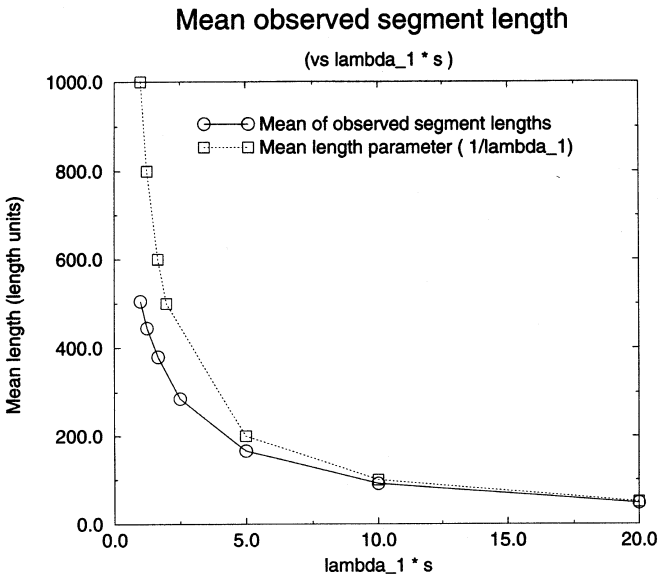
Our first step is to validate that the distribution of difference in the angle estimates for adjacent pixels can be approximated by a Von-Mises distribution with zero mean and precision parameter $\kappa/2$ (where κ is the precision parameter of each individual angle estimate). Figure 2 illustrates the histograms (binsize = 0.05 radians) for the difference angle estimate obtained over 1000 trials for $g/\sigma = 3.0$. The expected precision parameter for the individual angle estimates is approximately equal to $(g^2/\sigma^2) * f(w)$. The factor $f(w)$ here is a scale factor that is a function of the neighborhood size, w, used for the estimation of row and column gradients. $f(w)$ is equal to 50 for the 5 by 5 kernel. The predicted precision parameter for the difference angle estimate is approximately 225.0. The estimated precision parameter is 229.172. While this validates the theory, one must note that the approximation provided is valid only for large κ's. In addition, the approximation assumes independence between angle estimates for adjacent pixels. In reality, these angles are actually dependent because of the overlap in neighborhood windows. When the dependence is considered, then the angle-difference distribution still has mean zero, but a reduced variance (or in other words, increased precision parameter value). The estimated precision parameter for this case is found to be 294.069.



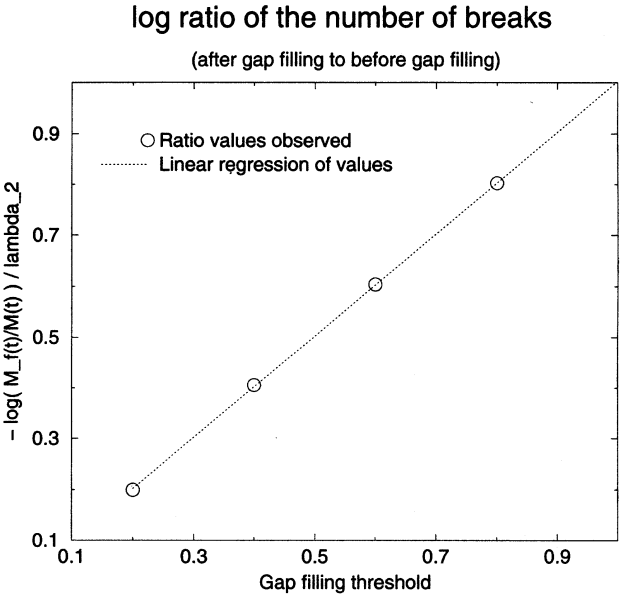
4



6



5



7

Fig. 4. Plot of mean number of breaks (with corresponding standard deviation estimates from the probability mass functions in previous figure) versus $\lambda_1 * 1000$. The theoretical approximation for large λ_2's is the linear function plotted

Fig. 5. Plot of observed mean segment length versus $(\lambda_1 * s)$ for a boundary of length s = 1000.0 units. Note how the mean of lengths of the segments observed deviates from the mean parameter of the renewal process (that governs the segment lengths) when $(s * \lambda_1) > 5$. This is the instance at which the truncation of the process at a given length s has significant effect on the mean parameter of the observed segment lengths. It is important to note that the mean parameter $1/\lambda_1$ and the mean length of the observed segments are not the same

Fig. 6. Plot of observed mean gap length versus $(1/\lambda_1)$ for a boundary of length s = 1000.0 units. Note that in this case, since $(1/\lambda_2) << s$ the observed gap lengths are very close to the mean parameter of the renewal process $1/\lambda_2$. As in the case of the segment length distribution, when $1/\lambda_2$ is comparable to s, we see the effects of truncation of the renewal process at length s

Fig. 7. Plot of $r(T_g) = -log(M_f(s)/M(s))/\lambda_2$ vs gap-filling threshold T_g for a boundary of length s = 1000.0 units. The observed values $r(T_g)$ should be equal to T_g

7.3 Validation of continuous-domain results

To validate the theoretical results we used 5000 trials to generate instances in the renewal process and gathered the empirical distributions for the quantities of interest outlined above (i.e., segment length, gap length, number of breaks, etc.). The parameters governing the renewal process λ_1, λ_2, L (the gap filling threshold) were varied. Figure 3 illustrates the probability mass function for the number of breaks in an interval of length s = 1000.0 units. Figure 4 illustrates

the relationship between the mean number of breaks versus $\lambda_1 * s$. As expected, the mean number of breaks is a linear function of the interval length s and the parameter λ_1. Figure 5 illustrates how the mean of the observed segment lengths varies as a function of $\lambda_1 * s$. It can be seen that when $s * \lambda_1 > 5$, the observed mean is not the same as the mean segment length parameter $1/\lambda_1$ of the renewal process. This is expected since the process is truncated at length s. In fact, we should view the segment and gap length distributions as being parametrized by three parameters λ_1, λ_2 and s. Figure 6 illustrates how the mean gap length varies as a function of $1/\lambda_1$. Since the gaps are very small in length compared to the total length of the interval (s) the mean of the observed gap lengths is close to the renewal process's original mean parameter for gaps ($1/\lambda_2$). Figure 7 illustrates the validation of the analytical result relating the mean number of breaks after the gap-filling operation to the mean number of breaks in the input process and the gap-filling threshold.

7.4 Discussion

Even though we have validated the correctness of the analytical results provided in the paper, there are, however, several issues that are not addressed here. First, in order to do the theoretical analysis, the estimated parameters at each pixel, i.e., gradient magnitudes, angle, were assumed to be independent of each other. This assumption, although correct for the case where the neighborhoods centered on the pixels do not share any common point, is a limiting one. Possible ways of handling this is discussed in [13]. We did not discuss here how the probabilities of false alarm p_f and misdetection p_m can be obtained since this is the subject of the section on edge-detector performance characterization in [13]. In fact, the p_m and p_f are functions of the neighborhood operator size, the gradient magnitude threshold T, and the signal-to-noise ratio (g/σ) (the ratio of the true gradient magnitude to the standard deviation of noise in the input). The parameter λ_1 is given by $p_m(T, g, \sigma, K)/(1.0 - p_m(T, g, \sigma, K))$. Note the usage of the notation $p_m()$ by recognizing that $p_m()$ is a function of the input parameters in the edge-detection step. It is also to be emphasized that the derivations in this paper assume stationarity of the sequence of binary random variables. Moreover the derivations for the length distributions are pertinent only when questions are asked relative to a a randomly chosen observed boundary fragment that is part of a perturbed boundary. A slightly different question is: Given an ideal boundary and a process that fragments the boundary into pieces, what is the likelihood of observing n pieces, $b_1, \ldots, b_n$, with lengths $l_1, \ldots, l_n$ and gaps $g_1, \ldots, g_{n-1}$? This question is addressed in a different paper by Haralick et al. [7]. Haralick et al. provide an expression for the above likelihood and analyze the effect of the morphological dilation operation on an observation of a random binary sequence.

8 Conclusion

In this paper we illustrated how one could set up random perturbation models for an example vision sequence involving edge finding, linking, and gap filling. This paper discussed how it is possible to propagate random perturbation models successively through the sequence. This paper does not discuss how one could utilize the random perturbation models for automatic selection of free parameters of the vision algorithms. This is the subject of the paper [14]. Theoretical results concerning the boundary error model were derived. Theoretical results are validated through systematic experiments. It was assumed here that the true gradients g_i are independent samples from a prior distribution. The nature of the prior distribution and the mathematical equations describing the probability density function vary with application domain. This assumption made the analysis a little simpler. In general, the true gray values at a particular pixel may depend on object characteristics, illumination direction, sensor position, etc. Thus, expecting the g_i's to be independent samples is not necessarily meaningful. It may often be the case that if we know something about the gradient at one location we may be able to say something about the gradient magnitude a few pixels away. This will make the analysis more involved.

Acknowledgements. The work described here was possible in part from a IBM Manufacturing Research Fellowship and a research grant from ARPA (contract 92-F1428000-000).

References

1. Abramowitz M, Stegun IA: Handbook of mathematical functions, National Bureau of Standards Applied Mathematics Series, 55
2. Haralick RM (1989) Performance assessment of near perfect machines, Machine Vision Appl, 2: 1–16
3. Burns B, Hanson A, Riseman E (1984) Extracting straight lines, COINS technical report, 84-19, December 1984
4. Haralick RM (1980) Edge and region analysis for digital image data. Comput Graph Image Process 12: 60–73
5. Haralick RM, Shapiro LG (1991) Computer and robot vision. Reading, Mass, Addison-Wesley
6. Haralick RM (1992) Performance characterization in image analysis: thinning, a case in point. Pattern Recogn Lett 13: 5–12
7. Haralick RM, et al. (1993) Statistical morphology. In: Mardia KV (ed) Advances in statistical image analysis. Also in: Proceedings of SPIE Conference on Image Algebra and Mathematical Morphology, vol 2030, pp 191–202, 1993
8. 'Dialogue: performance characterization in computer vision (1994) With contributions from Haralick RM, Cinque L, Guerra C, Levialdi S, Weng J, Huang TS, Meer P, Shirai Y, Draper BA, Beveridge JR. CVGIP: Image Understand 60: 245–265
9. Haralick R (1994) Performance characterization protocol in computer vision. ARPA Image Understanding Workshop 1994 Proceedings, vol 1, Morgan Kaufman Publishers, pp 667–674.
10. Johnson NL, Kotz S (1970) Continuous univariate distributions, vol 1, 2. Wiley, New York
11. Mardia KV (1972) Statistics of directional data, Academic Press, New York
12. Press SJ: Linear combinations of non-central chi-square variates. Ann Math Stat 37: 480–487
13. Ramesh V (1995) Performance characterization of image understanding algorithms. Ph.D. Dissertation, University of Washington, Seattle
14. Ramesh V, Haralick RM, (1994) A methodology for automatic selection of IU algorithm tuning parameters. Proceedings of the ARPA Image Understanding Workshop vol 1, pp 675–688, Morgan Kaufman, Hove

15. Ramesh V, Haralick RM (1992) Performance evaluation of edge operators. Presented in the special session on Performance Evaluation of Modern Edge Operators, Orlando, SPIE Machine Vision and Robotics Conference, held 20–24 April 1992
16. Stoyan D, Kendall WS, Mecke J (1987) Stochastic geometry and its applications. John Wiley, New York

Visvanathan Ramesh obtained his Bachelors in Electronics and Communcation Engineering with honors from Anna University Guindy, Madras, India, in 1984. He obtained his Masters degree in Electrical Engineering from Virginia Polytechnic and State University, Blacksburg, VA, in 1987 and defended his dissertation from the University of Washington, Seattle in December 1994. Currently, he is a member of technical staff at Siemens Corporate Research in Princeton NJ. He has been actively involved in Image Understanding research in the areas of low and mid level vision. His current goal is to build robust IU systems and to quantify robustness of IU algorithms. He has also focused on the research and development of software environments for Computer Vision. Besides his involvement in the Unix version of GIPSY (a general image processing system), he is a member of the ARPA Image Understanding Environment Committee, the committee that designed the IUE (an object oriented environment for Image Understanding Research). Dr. Visvanathan Ramesh can be contacted through email at rameshv@scr.siemens.com.

Robert M. Haralick received from the University of Kansas a BA degree in mathematics in 1964, a BS degree in electrical engineering in 1966, and a MS degree in electrical engineering in 1967. In 1969, after completing his PhD at the University of Kansas, he joined the faculty of the Electrical Engineering Department where he served as Professor from 1975 to 1978. In 1979, Dr. Haralick joined the Electrical Engineering Department at Virginia Polytechnic Institute and State University where he was a Professor and Director of the Spatial Data Analysis Laboratory. From 1984 to 1986, Dr. Haralick served as Vice President of Research at Machine Vision International, Ann Arbor, Michigan. Dr. Haralick now occupies the Boeing Clairmont Egtvedt Professorship in the Department of Electrical Engineering at the University of Wahsington. Professor Haralick was elected Fellow of IEEE for his contributions in computer vision and image processing. He serves on the Editorial Boards of Machine Vision and Applications and Real Time Imaging, and he is an Associate Editor for IEEE Transactions on Image Processing and Journal of Electronic Imaging. Professor Haralick's recent work is in shape analysis and extraction using the techniques of mathematical morphology, and in robust pose estimation, techniques for making geometric inferences from perspective projection information, propagation of random perturbations through image analysis algorithms, and document analysis. He has developed the morphological sampling theorem that establishes a sound shape/size basis for the focus of attention mechanisms that can process image data in multiresolution mode, thereby making some of image feature extraction processes execute more efficiently. Professor Haralick can be contacted by email at haralick@ee.washington.edu.