Random Perturbation Models & Performance Characterization in Computer Vision

Visvanathan Ramesh* and Robert M. Haralick
Department of Electrical Engineering, FT-10
University of Washington
Seattle WA 98195, U.S.A

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. These random perturbation models are useful for performing model based theoretical comparisons of the performance of vision algorithms. Parameters of these random perturbation models are related to measures of error such as the probability of misdetection of feature units, probability of false alarm, and the probability of incorrect grouping. Since the parameters of the perturbation model at the output of an algorithm are indicators of the performance of the algorithm, one could utilize these models to automate the selection of various free parameters (thresholds) of the algorithm.

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, there is no performance characterization which has been established in the research literature. What does performance characterization mean for an algorithm which might be used in a machine vision system? The algorithm is designed to accomplish a specific task. If the input data is perfect and has no noise and no random variation, the output produced by the algorithm ought also to be perfect. Otherwise, there is something wrong with the algorithm. So 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 which 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 which specifies how the imperfect perturbed data arises from the perfect data. Finally, we need a criterion function which 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 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. Due to fact that there are two types of errors, misdetection and false alarm, the output data consists 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: 1) perturbation models for perturbed true feature entities in the output, and 2) perturbation models for purely random feature entities that appear in the output. The first part

*The support by IBM through an IBM manufacturing research fellowship is gratefully acknowledged.

0-8186-2855-3/92 $3.00 ©1992 IEEE

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 are related to measures of error such as the probability of misdetection of feature units, probability of false alarm, and the probability of incorrect grouping. Since the parameters of the perturbation model at the output of an algorithm is an indicator of the performance of the algorithm, one could utilize these models to automate the selection of various free parameters (thresholds) of the algorithm. We organize this paper into two parts, 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. We then provide theoretical plots of performance measures at each stage of our vision algorithm sequence.

2 Random Perturbation Models for Boundary Extraction Sequence

In this section we discuss a theoretical model by which pixel noise can be sucessively propagated through an edge labeling algorithm, an edge linking algorithm, and a boundary gap filling algorithm. We concentrate on describing the idealizations for the spatial configurations of interest and the nature of the random perturbations which affect these idealizations at each stage.

2.1 Perturbation model at input of edge detector/linker

Our edge idealization is that of a linear ramp edge. Our random perturbation model is that each pixel value is corrupted with additive zero mean Gaussian noise with known standard deviation. Due to the noise, the edge labeling process mislabels some true edge pixels as non-edge pixels. We call these misdetected edge pixels. Our non-edge idealization is that of a flat graytone surface. Our random perturbation model is that each pixel value is corrupted with additive zero mean Gaussian noise with known standard deviation. Due to the noise, the edge labelling process mislabels some true non-edge pixels as edge pixels.

2.2 Edge detection step – Analysis

There are two kinds of errors in the feature extraction step, namely: misdetection of true feature entities and false labelling of non-feature entities. Two measures of performance of an edge detector are the probability of misdetection and the probability of false alarm. In this section we derive expressions for the probability of misdetection and probability of false alarm of gradient based edge operators.

2.2.1 Probability of Misdetection of a Gradient Edge

We assume that the gradient at a particular pixel is estimated by computing an equally weighted least squares fit to the gray levels in the pixel's neighborhood. It is also assumed that the input image is corrupted with additive Gaussian noise with zero mean and variance σ^2. If we approximate the image graytone values in the pixel's neighborhood by a plane $\alpha r+\beta c+\gamma$, then the gradient value $g = \sqrt{\alpha^2+\beta^2}$. To estimate α and β we use a least squares criterion. On the basis of these estimate, we can derive the density function for the estimated gradient.

Under our assumptions about the noise model in the input image it can be shown that the fitted parameters $\hat{\alpha}$ and $\hat{\beta}$ are Gaussian random variables with means μ_α, μ_β and variances $\sigma_\alpha^2, \sigma_\beta^2$ respectively. We use the notation U_i to denote unit normal random variables with zero mean and unit variance. Under this notation we can rewrite the expressions for α and β as: $\alpha = \mu_\alpha + \sigma_\alpha U_1$ and $\beta = \mu_\beta + \sigma_\beta U_2$. Note that: ${\sigma_\alpha}^2 = {\sigma_\beta}^2$ when a square neighborhood is used in the fit and they are related to the input noise variance σ^2 by the expression:${\sigma_\alpha}^2 = \frac{\sigma^2}{\Sigma_r\Sigma_c r^2}$. Here $\Sigma_r\Sigma_c r^2$ is the summation of the squared row index values over the neighborhood used in the least squares fit.

$\sum_i^v (U_i+\delta_i)^2$ is distributed as a non-central chi-square distribution with non-centrality parameter $\sum_{i=1}^v \delta_i^2$. Now:

$$\hat{G}^2 = \hat{\alpha}^2 + \hat{\beta}^2 = \sigma_\alpha^2 . {\chi_1}^2(\delta_\alpha^2) + \sigma_\beta^2 . {\chi_1}^2(\delta_\beta^2) \quad (1)$$

where: $\delta_\alpha = \frac{\mu_\alpha}{\sigma_\alpha}$ and $\delta_\beta = \frac{\mu_\beta}{\sigma_\beta}$. Now, $(U_1+\delta_\alpha)^2$ and $(U_2+\delta_\beta)^2$ are non-central chi-square distributed with noncentrality parameters δ_α^2 and δ_β^2 and 1 degree of freedom. The distribution of the sum of two non-central chi-square distributed random variables is also non-central chi-square distributed. Press [3] has shown that the distribution of linear functions of independent non-central chi-square variates with positive coefficients can be expressed as mixtures of distributions of central chi-square's. In the situation where the input noise is additive zero mean Gaussian noise we have shown that the ratio $G^2/\sigma_\alpha{}^2$ is a non-central chi-square distribution. with 2 degrees of freedom and non-centrality parameter $C = (\mu_\alpha{}^2 + \mu_\beta{}^2)/\sigma_\alpha{}^2$. The probability of misdetecting the edge is given by

$$q = Prob\left(\chi_2{}^2(C) < \frac{T^2 \Sigma_r \Sigma_c r^2}{\sigma^2}\right) \qquad (2)$$

2.3 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 an edge linking scheme is the probability of correct grouping of edge pixels. In the next section we show how the distribution for the edge orientation can be approximated by a Von Mises distribution. Using this result, we then derive the expression for the probability of correct grouping of true edge pixels.

2.3.1 Probability of correct edge grouping

The gradient magnitude is given by $\sqrt{\hat{\alpha}^2 + \hat{\beta}^2}$. The estimated edge direction is given by: $\theta = tan^{-1}(\hat{\beta}/\hat{\alpha})$. Now, $\hat{\alpha}$ and $\hat{\beta}$ are Gaussian random variables with parameters $(\mu_\alpha, \sigma_\alpha)$ and $(\mu_\beta, \sigma_\beta)$ respectively. Hence $tan\theta$ is distributed as the quotient of two non-standardized normal random variables. Springer [5] gives the derivation of the probability density function for this case. Mardia [2] gives the derivation for the probability density function for the orientation (by letting $\hat{\alpha} = g\cos\theta$ and $\hat{\beta} = g\sin\theta$) in terms of the bivariate normal distribution for (α, β). The distribution so derived is called the offset normal distribution and is related to the Von Mises distribution which is widely used in the literature. The Von Mises distribution can be obtained from the offset normal distribution if we assume equal variances for $\hat{\alpha}$ and $\hat{\beta}$. Indeed, as shown in Haralick [1], the estimates for the variances of $\hat{\alpha}$ and $\hat{\beta}$ will be the same for square facet windows. Hence in the discussion that follows we assume that Θ has the Von Mises distribution with parameters $\mu_0 = tan^{-1}(\hat{\beta}/\hat{\alpha})$. and $\kappa = 1/\sigma_\alpha{}^2$. A random variable Θ is said to be Von Mises distributed if:

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

$$\kappa > 0, \quad 0 \le \mu_0 < 2\pi$$

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:

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

In the case of an edge linker which 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 $(\theta_1 - \theta_2) mod 2\pi$ is small enough, then the pixels are linked. To determine what is 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 $(\theta_1 - \theta_2) mod 2\pi$ is derived in Mardia [2]. It is shown in [2] 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:$A(\kappa_3) = A(\kappa_1)A(\kappa_2)$

$$A(x) = 1 - \frac{1}{2x} - \frac{1}{8x^2} - \frac{1}{8x^3} + o(x^{-3}) \qquad (5)$$

When $\mu_1 = \mu_2 = \mu$ and $\kappa_1 = \kappa_2 = \kappa$ then $\mu_3 = 0$ and $\kappa_3 \approx \frac{\kappa}{2}$, 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.

2.4 Perturbation model at edge detector/linker output – Misdetection

In the above discussion, we derived expressions for various measures of performance of an edge detector and linker. In this section we show how these measures relate to an algorithm employed at a subsequent stage, such as the gap filling algorithm. We now illustrate how the boundary output obtained can be visualized as breakages of the ideal expected boundary, broken by a renewal process. In addition we illustrate how the various probabilities calculated at the previous sections 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: $P_{segmentlength}(l = k) = p^k q$. This is the geometric distribution which is a special case of the negative binomial distribution. The distribution of gap lengths between two edge segments is given by: $P_{gaplength}(l' = k) = q^k p$. 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:

$$P_{segmentlength}(l = k) = \frac{(p^k q)}{1 - p^{L+1}}. \qquad (6)$$

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

$$P_{gaplength}(l' = k) = \frac{(q^k p)}{1 - q^{L+1}}. \qquad (7)$$

2.5 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 inter-event 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.

2.5.1 Probability density function of interval between two breaks

We know that the renewal process consists of alternating edge segments and gaps. The edge segment lengths and the gap lengths are both 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: $Prob(X = x) = \lambda exp^{-\lambda x}$. The exponential distribution is the continuous analog of the geometric disribution. The parameter, λ, of the exponential distribution can be shown to be equal to the ratio (q/p) of the parameters in the geometric distribution. 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. 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:

$$Prob(X = x) = \frac{\lambda_1 \lambda_2}{\lambda_2 - \lambda_1}\left(e^{-\lambda_1 x} - e^{-\lambda_2 x}\right) \qquad (8)$$

2.5.2 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 equation (8). Let $Y_1, Y_2, \ldots, Y_M$ be i.i.d random variables with probability density function given by equation(8). Let $N(t)$ denote a counting process which gives the number of breaks in an interval t. The process generated is a renewal process with probability density function for each interval being:

$$Prob(Y = y) = \frac{\lambda_1 \lambda_2}{\lambda_2 - \lambda_1}\left(e^{-\lambda_1 y} - e^{-\lambda_2 y}\right) \qquad (9)$$

The expected number of events in an interval of t

is given by: $M(t) = \overline{N(t)} = \sum_{k=0}^{\infty} k Prob(N(t) = k)$ It can be shown that:

$$M(t) = \left(\frac{\lambda_1\lambda_2}{\lambda_1+\lambda_2}\right)\left[t - \frac{1-e^{-(\lambda_1+\lambda_2)t}}{\lambda_1+\lambda_2}\right] \quad (10)$$

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 $\lambda_1 << \lambda_2$ and $1 >> e^{-\lambda_2 t}$ and the expression for $M(t)$ becomes: $M(t) = \lambda_1 t$. That is, the mean number of breaks in the true boundary is a linear function of the length of the boundary.

2.6 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. Due to lack of space we do not include the results for the distributions of the segment and gap lengths.

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: $p(1-p)^{i-1}$. Using the fact that the output process intervals are obtained by random convolution of the interval times in the input process, we can show that the expression for the mean number of breaks in the output of the gap filling procedure is given by:

$$M_f(t) = \left(\frac{\lambda_1\lambda_2 p}{(\lambda_1+\lambda_2)^2}\right)\left[(\lambda_1+\lambda_2)t - (1-e^{-(\lambda_1+\lambda_2)t})\right] \quad (11)$$

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

$$M_f(t) \simeq M(t)p = M(t)e^{-\lambda_2 L}. \quad (12)$$

This means that the mean number of gaps in the output is the product of the expected number of gaps in the input, $\lambda_1 t$, and the probability that a gap is not filled, p.

2.7 Perturbation Models for Random Entities

In the above analysis we modelled the perturbations to the true features we are looking for, namely curve/line segments. Here we now focus on modelling false features that are produced in various stages of the algorithm sequence. At the edge detection step the input image gray values are assumed to be corrupted with noise which may be modelled as a Gaussian distribution with zero mean and standard variance σ. That is: $I(r,c) = I_t(r,c) + \eta(r,c)$, where, $I(r,c)$ is the observed image gray value, $I_t(r,c)$ is the true gray value and $\eta(r,c)$ is the noise component. Often $\eta(r,c)$ is assumed to be zero mean Gaussian with a standard deviation of σ. If we had correlated noise in the gray level image, the edge detector labels non-edge pixels as edge pixels and the edge linking and fitting step produces short line segments at the output. We first illustrate how the noise model in the input propagates to the output of the edge operator. We model the process seen at the input of the edge linker as a random point process.

2.7.1 Probability of False alarm at the edge detector output

We assume that the input data at the edge detection step is a region of constant gray tone values with additive Gaussian noise. Since a pixel is labelled an edge pixel if the estimated gradient value, G, is greater than a specified threshold, T, the probability of false detection is $P(G > T)$. The coefficients α and β of the facet model described in chapter 2 are normally distributed with zero mean. If the input noise variance is σ^2 then the variance of α, ${\sigma_\alpha}^2$ is equal to: $\sigma^2/\sum_r\sum_c r^2$. The variance of β, ${\sigma_\beta}^2$, is equal to: $\sigma^2/\sum_r\sum_c c^2$.

Note that the summations are done over the index set for r and c. Since $G^2 = \alpha^2 + \beta^2$, if we assume a square neighborhood then the $G^2/{\sigma_\alpha}^2$ is chi-square distributed with 2 degrees of freedom. So the probability of labelling of a noise pixel as an edge pixel can be computed once we know the variance for the parameter α. Note that only when the operator uses a square neighborhood the estimates of the variances for α and β are equal. The above simplification is possible only under this condition. On the other hand when a rectangular neighborhood is used the only difference is that G^2 is distributed as a linear combination of two chi-square distributed random variables.

2.7.2 Noise process at the input of edge linker

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, the spatial pattern generated follows a discrete random process. The discrete random process we use is the *Bernoulli lattice process*, see [6]. The Bernoulli lattice process is the discretized analogue of the Poisson point process.

2.7.3 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 gets 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 general solution is given in [6]. Let $D(r)$ denote the probability distribution function of the nearest neighbor distance, then $D(r) = 1 - e^{-(\lambda \pi r^2)}$. The mean distance can be shown to be: $\mu_r = \frac{1}{2\sqrt{\lambda}}$. 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 there exists no edge pixel 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. However, the gap filling algorithm fills the gaps and therefore produces segments instead of points. The process so obtained is a line segment process.

3 Theoretical Plots

In this section we provide theoretical plots for some of the performance measures derived in the previous sections. Specifically, we plot the theoretical false alarm and misdetection characteristics of gradient based edge detectors. Figure 1 gives the false alarm vs misdetection curve for a gradient based edge detector that uses equally weighted least squares fit (5 by 5 window size, Noise variance = 25). Figure 2 gives the theoretical plot of the probability of misdetection versus squared gradient threshold for various edge slopes. Figure 3 shows the probability of false alarm versus squared gradient threshold, for various noise variances and a true edge slope of 4.0. Figure 4 gives the mean gap length between true edge segments as a function of squared gradient threshold. A true boundary segment of length 100 pixels was assumed and the noise variance was 25.0, window size = 5 by 5, edge slopes of 2.0, 3.0, and 4.0. It can be seen from this plot that the mean gap length (for the most part) varies linearly with the squared gradient threshold. Figure 5 is a plot of the mean lengths of the random short segments (false alarm) as a function of squared gradient threshold. It can be seen that the segment lengths are rather small, this is due to the fact that the plot does not account for the gap filling step.

4 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 discusses how the models fit together for the example sequence and does not provide comparisons regarding the superiority of one technique over the other. Following the same methodology, a comparison of the performance of gradient based edge operators and morphogical edge operators is done in [4].

References

[1] Haralick, R.M., *"Edge and Region Analysis for Digital Image Data,"* Computer Graphics and Image Processing, Vol.12, 1980, pp. 60-73.

[2] Mardia, K.V., *Statistics of directional data*, New York: Academic Press, 1972.

[3] Press, S.J., *"Linear combinations of non-central chi-square variates,"* Annals of Mathematical Statistics, Vol. 37, pp. 480-487.

[4] V.Ramesh, R.M.Haralick, *"Performance evaluation of edge operators,"* Submitted to the special session on Performance Evaluation of Modern Edge operators, Orlando Machine Vision and Robotics Conference to be held 20-24 April 1992.

[5] Springer, M.D., *The Algebra of Random Variables*, New York: John Wiley and Sons, 1979.

[6] D.Stoyan, W.S.Kendall, J.Mecke, *Stochastic Geometry and its Applications*, John Wiley and Sons, 1987.

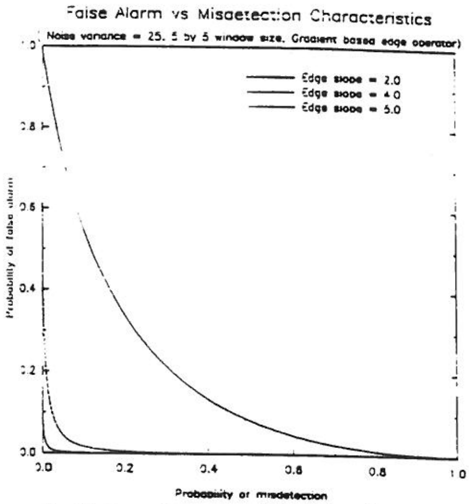

Figure 1: False alarm vs Misdetection curve

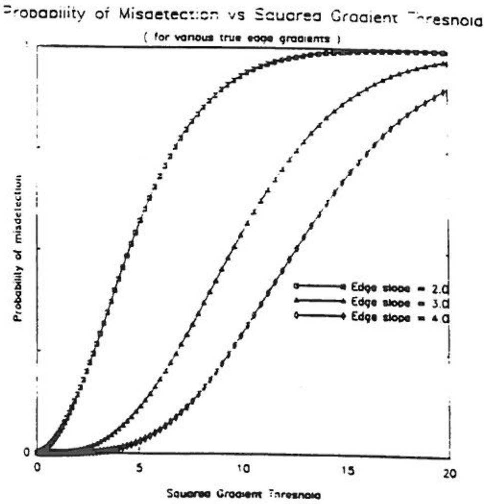

Figure 2: Probability of Misdetection vs Squared Gradient Threshold

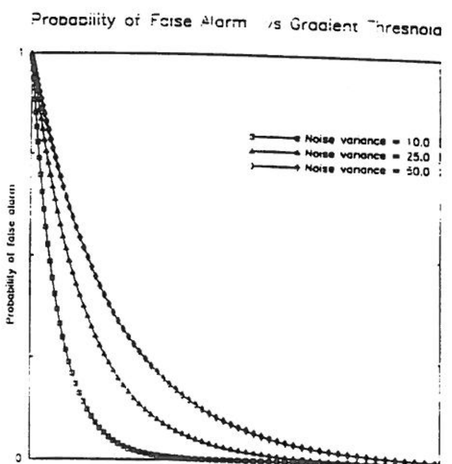

Figure 3: Probability of False alarm vs Squared Gradient Threshold

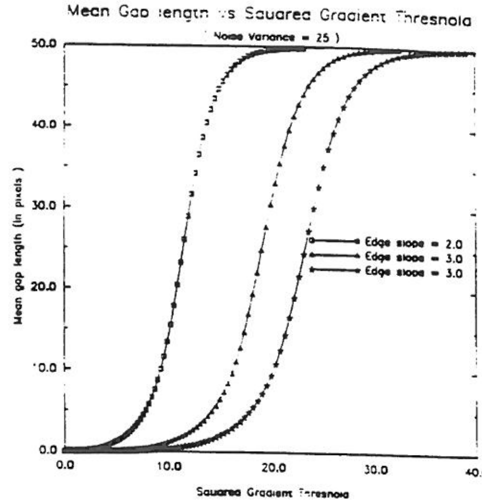

Figure 4: Mean gap length vs Squared Gradient Threshold

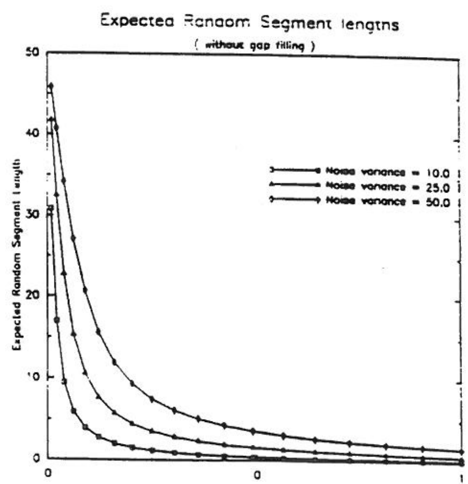

Figure 5: Mean random segment length vs Squared Gradient Threshold