Performance Characterization of Edge Operators

Visvanathan Ramesh [*]
Robert M. Haralick
Department of EE, FT-10
University of Washington
Seattle WA 98195

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 the paper by Ramesh and Haralick, [1], theoretical models for the random perturbations in the output of a vision sequence, involving edge finding, edge linking and line fitting were presented. They modelled the process that describes the breakage of a true model line segment by a renewal process with alternating line and gap intervals. However, their paper assumed independence of gradient estimates obtained from neighboring pixel locations.

In this paper we show how one can relax the independence assumptions and derive perturbation models that include the effects of correlation between neighboring gradient estimates. Under the assumption that the ideal data is corrupted with additive, independent additive gaussian noise, we derive expressions that describe the relationship between an edge gradient estimate at a given location and an edge gradient estimate for a neighboring pixel. We illustrate how the model line breakage process can be modeled as a Markov process whose parameters are functions of the true edge gradient, edge operator's neighborhood size, and the noise variance. Furthermore, we derive theoretical expressions for the mean positional error as a function of the neighborhood operator window size, noise variance, the width of the true ramp edge, and the true edge gradient. We also outline an experimental protocol used for evaluating edge pixel positioning errors and discuss the results obtained from the experiments.

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 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 the paper by Ramesh and Haralick, [1], theoretical models for the random perturbations in the output of a vision sequence, involving edge finding, edge linking and line fitting were presented. They modelled the process that describes the breakage of a true model line segment by a renewal process with alternating line and gap intervals. However, their paper assumed independence of gradient estimates obtained from neighboring pixel locations.

In this paper we show how one can relax the independence assumptions and derive perturbation models that include the effects of correlation between neighboring gradient estimates. Under

[*] This project has been funded in part through a contract from DARPA. Funding for V.Ramesh from IBM in the form of a IBM Manufacturing Research Fellowship is also gratefully acknowledged

the assumption that the ideal data is corrupted with additive, independent additive gaussian noise, we derive expressions that describe the relationship between an edge gradient estimate at a given location and an edge gradient estimate for a neighboring pixel. We illustrate how the model line breakage process can be modeled as a Markov process whose parameters are functions of the true edge gradient, edge operator's neighborhood size, and the noise variance. Furthermore, we derive theoretical expressions for the mean positional error as a function of the neighborhood operator window size, noise variance, the width of the true ramp edge, and the true edge gradient. This paper is organized as follows. The first section provides a brief review of results discussed in [1]. The second section provides an analysis of the positional error introduced by gradient based edge operators. The third section provides a discussion of a perturbation model for the edge output that takes into account the dependence of estimates at neighboring pixels. A subsequent section provides a discussion of theoretical performance measure plots.

2 Review of previous results

In this section we review some of the results outlined in the paper by Ramesh and Haralick [1]. Our work extends the results given in [1]. Ramesh and Haralick [1], describe a theoretical model by which pixel noise can be successively propagated through an edge labelling algorithm, an edge linking algorithm and a boundary gap filling algorithm. Assuming an edge idealization of a linear ramp edge and i.i.d Gaussian random perturbations on pixel grayvalues they show how one could model the breakage of a true line segment as a renewal process with alternating segment and gap intervals. They show that if one ignores the dependencies between adjacent gradient estimates then the segment and gap interval lengths are exponentially distributed with parameters λ_1 and λ_2 that are related to the true edge gradient, the neighborhood operator size and the gradient threshold employed. They also show how the output after a gap filling operation could still be modeled as an alternating renewal process and derive the length distributions for the segment and gap intervals after the operation.

In reality, there is an overlap between the edge detector neighborhoods centered around pixels and hence there is some dependence between gradient estimates obtained for neighboring windows. In addition, if one assumes that the noise at each pixel is locally dependent then the correlation in the noise would introduce correlation in the gradient estimates. In addition, the analysis in [1] did not include positional errors. These positional errors are of significance if one wishes to analyze higher-level matching algorithms.

In other work, [2], we focussed on performing theoretical model-based comparison of gradient based edge finding schemes and mathematical morphology based edge finding schemes. The performance analysis was done by assuming an ideal edge model and a noise model and by deriving expressions for probability of false alarm and probability of misdetection of edge pixels. Under the Gaussian noise model assumption, the theory indicated that the morphological edge detector is superior to conventional gradient based edge detectors, that label edges based on gradient magnitude, when a size 3 by 3 window was used. We performed experiments to validate our theoretical results and the empirical plots indicated that the morphological edge operator was also superior when a 5 by 5 window is used. However, the theoretical plots did not confirm this because the theory provided only an upperbound. In [2] we also included comparisons of results obtained for real images. A simple analysis of hysteresis linking was also done in this paper and it was shown that hysteresis linking improves the performance of the edge operators.

3 Positional Error Analysis of Gradient Based Edge Detectors

In this section we derive the expression for the mean error in the edge pixel location. We consider an ideal edge model that has an one-dimensional intensity profile of a ramp. Specifically, the intensity profile is defined by:

$$\begin{aligned} I(x) &= a + Gx \\ &\quad \text{for } x = -K - 1/2, \ldots, K - 1/2 \end{aligned} \tag{1}$$

$$\begin{aligned} &= a - G(K-1)/2 \text{ for } x < -K - 1/2 \\ &= a + G(K-1)/2 \text{ for } x > K - 1/2 \end{aligned}$$

We assume that the edge detection is performed by computing the gradient by fitting a planar surface to the grayscale values as in [1]. In 1-dimension this problem is equivalent to fitting a line to the data for each 1 by K neighborhood. There are two kinds of errors that are introduced in the fit, one error is the systematic bias that is introduced in the fit due to the approximation of the function $I(x)$ by a linear fit in the 1 by K neighborhood and the other error is the error introduced due to the additive noise in the input. Let $G(x)$ be the gradient estimate obtained when the least squares fit is performed for the window of ideal data $I(i), i = x - (K-1)/2, \ldots, x + (K-1)/2$. Clearly, $G(0)$, the gradient estimate when the neighborhood overlaps the entire ramp, is equal to the true slope G. Also, $G(x), |x| > K$ is equal to zero. In addition, one can note that $G(x)$ is a symmetric function since $I(x)$ is symmetric. When the discrete samples are corrupted with additive i.i.d Gaussian noise with zero mean and variance σ^2, then the estimates for the gradient values, $\hat{G}(x)$, are normal random variables with true mean $G(x)$ and variance $\sigma^2/\sum i^2$ where the sum is taken over values of $i = -(K-1)/2, \ldots, (K-1)/2$. Neighboring gradient estimates, $\hat{G}(x)$ and $\hat{G}(x+j)$, are dependent random variables because of the overlap in the neighborhoods used during the estimation procedure.

We show in the appendix that if we viewed the sequence of $2K-1$ random variables $\hat{G}(x), x = -(K-1), \ldots, K-1$ as a random vector $\hat{\mathbf{G}}$ then $\hat{\mathbf{G}}$ is distributed as a multivariate Gaussian random variable with mean vector $\mathbf{G(x)}$ and covariance matrix $\mathbf{A\Sigma A'}$, where the matrix $\mathbf{A}$ is obtained from fitting kernel coefficients as described in the appendix and Σ is the covariance matrix of the additive noise vector which is assumed to be $\sigma^2\mathbf{I}$. The matrix $\mathbf{A}$ captures the dependence between the adjacent gradient estimates.

In order to compute the error in the edge pixel position, we assume that the pixel with the maximum gradient magnitude along the gradient direction is labelled as an edge, while all the other pixels are labelled as non-edge pixels. That is, the edge pixel's index is ep when:

$$\begin{aligned} &\hat{G}(ep) > \hat{G}(x) \\ &\forall x > -(K-1), x < (K-1), x \neq ep \end{aligned} \tag{2}$$

Hence the probability that the location i is labelled as edge is given by a multivariate integral with appropriate limits specified by the gradient threshold used. That is, the probability is given by the expression:

$$\begin{aligned} &Prob(ep = i) = \\ &\int_{x_i=T}^{\infty} \int_{\substack{x_j=0 \\ j\neq i}}^{x_i} \cdots \int \Phi(\mathbf{G(x)}, \mathbf{A\Sigma A'}) dx_i dx_j \\ &+ \int_{x_i=-\infty}^{-T} \int_{\substack{x_j=x_i \\ j\neq i}}^{0} \cdots \int \Phi(\mathbf{G(x)}, \mathbf{A\Sigma A'}) dx_i dx_j \end{aligned} \tag{3}$$

where Φ is the multivariate normal distribution function. Φ has two sums in the integral because the threshold T is actually on the absolute value of the gradient. The mean error in the edge pixel location is then given by:

$$\mu = \sum_{i=0}^{K-1} i \; Prob(ep = i) \tag{4}$$

4 Boundary Model incorporating Dependencies in Estimates

In the previous section we addressed how errors in grayvalues propagate to errors in pixel locations at the output of the edge operator. An alternating renewal process with gaps and edge segments ([1]) is used to describe the breakage of a true model segment into short edge segments and gaps. Under the assumption that the gradient across the edge is constant along the true model boundary and ignoring the dependence between gradient estimates from local neighbours, it was seen in [1] that the edge segment lengths and the gap lengths can be approximated as exponential distributions. In this section we illustrate how the boundary model given in [1] can be extended to include the dependencies due to correlation of gradient estimates.

We assume the ideal model for the intensity profile across the boundary to be a ramp edge with constant gradient as one walks along the model line. In addition we assume that the samples are corrupted with i.i.d. additive Gaussian noise. It is shown in the appendix how one can relate the gradient estimate obtained at a particular location, (r, c), to the gradient estimate at a nearby location, $(r+k, c+j)$. Using these relationships one can derive the expression for the probability that the gradient estimate $\hat{G}(r+k, c+j)$ is greater than the threshold T given that the gradient estimate $\hat{G}(r, c) > T$. For simplicity, we can assume that the ramp edge is oriented across the column direction. In this situation we are interested in modelling the relationship between the gradient estimates at successive rows. This scenario is equivalent to the examination of the gradient estimates in neighboring pixels, as one walks along the true model line. We visualize the sequence of edge labels (1's and 0's) as we walk along the model line as a series of binary random variables. It is easily seen that if we are dealing with independent Gaussian noise at each pixel, the gradient estimate $\hat{G}(r, c)$ is dependent on the previous $(\hat{G}(r-k, c), k = 1, \ldots, K)$ estimates. In this sense, the binary edge sequence forms a binary Kth order Markov chain. The Markov chain can be specified by the conditional probabilities: $Prob(X_r = 1 | X_{r-k} = 1)$, $k = 1, \ldots, K$ and $Prob(X_r = 1 | X_{r-1} = 1, \ldots, X_{r-j} = 1), j = 2, \ldots, K$. These probabilities can be easily derived from the joint distribution for the $\hat{G}()$'s by computing the appropriate multivariate integral. For example: $Prob(X_r = 1 | X_{r-k} = 1)$ is equal to $Prob(\hat{G}(r, c) \geq T | \hat{G}(r-k, c) \geq T)$ and is given by:$Prob(\hat{G}(r, c) \geq T, \hat{G}(r-k, c) \geq T) / Prob(\hat{G}(r-k, c) \geq T)$. The numerator in the above expression is obtained by integrating the joint distribution of $\hat{G}(r, c)$ and $\hat{G}(r-k, c)$ with limits of integration from T to ∞. The denominator is the integral of the distribution function for $\hat{G}(r-k, c)$.

5 Protocol for image generation (for edge pixel accuracy) and evaluation

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 $\rho = (r - R)cos(\theta) + (c - C)sin(\theta)$.

$$\begin{aligned} I(r, c) &= I_{min}, \quad \rho < 0 \\ &= I_{max}, \quad \text{otherwise.} \end{aligned} \tag{5}$$

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 degrees. To generate ramp edges, we averaged images containing the step edges with a kernel of size $4by4$ so that the resulting ramps have 5 pixels width. 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:

$$SNR = 20log\left(\frac{\sigma_s}{\sigma_n}\right) \tag{6}$$

where σ_s is the standard deviation of the gray values in the input image and σ_n is the noise standard deviation. We used SNR values of 0, 5, 10, 20 dB. They correspond to σ_s/σ_n values of 1, 1.78, 3.162, and 10 respectively. Groundtruth edge images were generated by using the following function where $\rho = (r - R)cos(\theta) + (c - C)sin(\theta)$.

$$\begin{aligned} I_1(r, c) &= 0 \quad \rho < -0.5 \\ &= 1 \quad \text{otherwise.} \\ I_2(r, c) &= 0 \quad \rho < 0.5 \\ &= 1 \quad \text{otherwise.} \\ I(r, c) &= I_1(r, c) \text{ exor } I_2(r, c) \end{aligned} \tag{7}$$

The operators employed included the gradient based (Gradient computed using the slope facet model) operator and the morphological blur-minimum operator discussed in [3]. In the Blur-minimum morphological edge detector a pixel is assigned an edge label if the edge strength computed is above a given threshold T. The edge strength I_e is given by the equation:

$$\begin{aligned} I_e = min\{I_1 - & \quad erosion(I_1, disk(r)), \\ & \quad dilation(I_1, disk(r)) - I_1\}. \end{aligned} \tag{8}$$

where I_1 is the input image and r is the radius of the disk that is used as the structuring element in the morphological erosion/dilation operations. We used 5 by 5 neighborhoods for the edge operator and the blur-minimum operator. The edge accuracy evaluation proceeded as follows. The edge pixel location error E is defined as the distance along the gradient direction from the true edge pixel to the nearest labelled edge pixel (if one exists, in the edge detector output). A given ground truth edge pixel is assumed to be missing in the detector output if if there are no edge pixels in the detector output within an interval centered on the ground truth edge pixel. The interval is oriented along the gradient direction and the number of pixels in the interval is equal to the edge operator width. We will refer to this interval, as the "valid zone" for each pixel.

In addition to the computation of edge pixel location error as given above, we also compute the following statistics from the output image. We visualize the edge and non-edge labellings encountered as one walks along the valid zone as a sequence of alternating 0 and 1 runs. We compute the mean and variances for the lengths of the gaps and the edge segments. In the ideal case when there is no error the edge segment lengths will have mean value of 1 and a variance of zero, whereas the gap segment lengths will have a mean value equal to the $\lfloor W/2 \rfloor$, where W is the window operator neighborhood size. At low levels of edge gradient threshold the edge detector responses are thick regions and the edge segment length values may vary from 1 to W. The segment length and gap length statistics capture this aspect. Given the true ground truth segment, the edge segment length and gap length statistics and a value for the probability of misdetection of the edge operator, we can generate a realization of the edge detector response by following the procedure outlined in the appendix.

6 Plots and Discussion

The results obtained from the experiments are given in figures 1 through 6. The curves were obtained by taking the running mean of adjacent samples. The window size for the running mean operation was 5. The results shown in the plots are the results obtained after 10 replications. Figures 1 and 4 illustrate how the mean length of the run of edge pixels varies with edge strength threshold for the morphological operator and the gradient based operator. It is clear from the plots that as the edge strength threshold is increased the run length drops to a value of 1. When the gradient threshold is high, we label lesser number of pixels as edge pixels in the output and hence the runs encountered are of small width. Another point that the plots illustrate is that as the signal to noise ratio increases from -5 to +20 dB the slope of the curve increases. This effect is due to the fact that the noise has the effect of smoothing on the ideal run-length profile. Ideally, we expect the run-length to be a linear function of the threshold (since the input consists of linear ramp edges). Figures 2 and 5 illustrate how the mean gap length varies with the edge strength threshold. As expected, the mean gap length monotonically increases as a function of the edge strength threshold. In the ideal case, we expect the mean gap length to be a linear function of the edge strength threshold and in the presence of large degree of noise this ideal function is blurred. Figures 3 and 6 illustrate how the mean edge pixel positional error varies with edge strength threshold. It is clear that the error drops to zero when the signal to noise ratio is high. When the signal to noise ratio is 0 or 5 dB it can be seen that the mean error is as much as 0.5 pixels. A comparison of the plots for the morphological and gradient based operators indicate that the gradient based scheme is superior for signal to noise levels of 0 dB and higher. The gradient based scheme has comparable errors when the signal to noise ratio is -5 dB. The conclusion in [2] was that the morphological operator had superior false alarm vs misdetect characteristics. The experiments here point out that the morphological operator has poorer localization performance. In a subsequent paper we will attempt to compare the empirical results obtained with theoretical results by utilizing the theoretical expressions for the mean pixel positioning error. The exact expression for the distribution of pixel error for the morphological edge operator discussed in [3] still needs to be worked out. We are also in the process of evaluating the noisy edge generation procedure, that utilizes similar statistics as in our experiments, to see how closely it models errors that occur in real images.

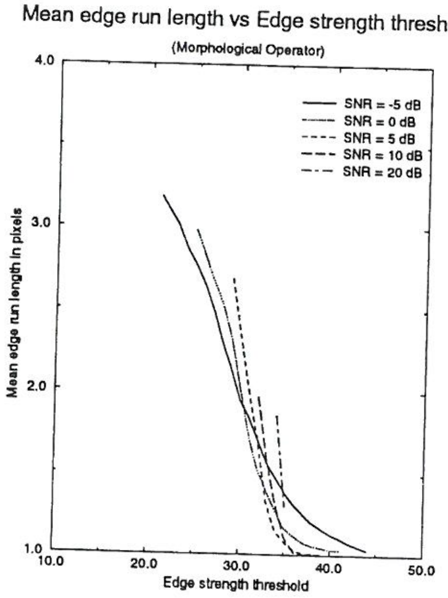

Figure 1: Plot of Mean edge run length vs Edge strength threshold for various signal to noise ratios. Orientation of the true edge was 15 degrees, Window size 5 by 5 for Morphological operator

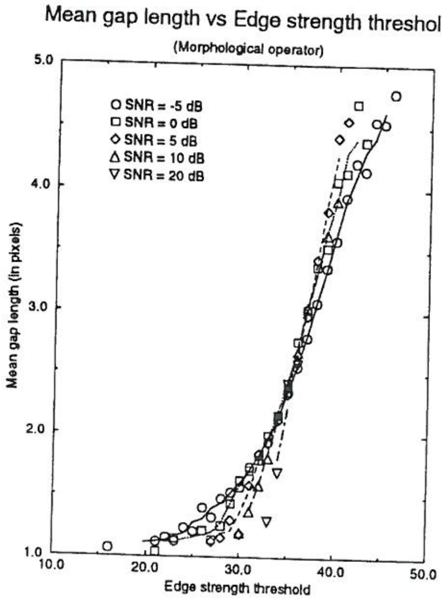

Figure 2: Plot of Mean gap run length vs Edge strength threshold for various signal to noise ratios. Orientation of the true edge was 15 degrees, Window size 5 by 5 for Morphological operator

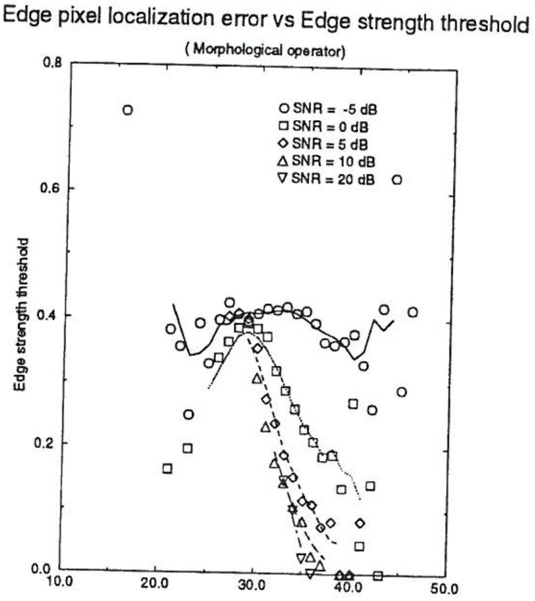

Figure 3: Plot of Mean pixel positional error vs Edge strength threshold for various signal to noise ratios. Orientation of the true edge was 15 degrees, Window size 5 by 5 for Morphological operator

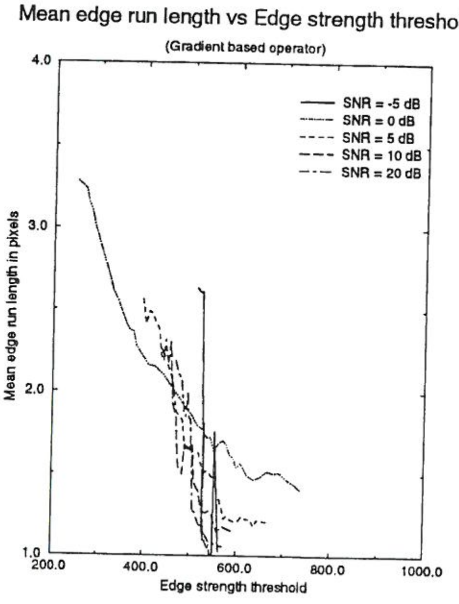

Figure 4: Plot of Mean edge run length vs Edge strength threshold for various signal to noise ratios. Orientation of the true edge was 15 degrees, Window size 5 by 5 for Gradient operator

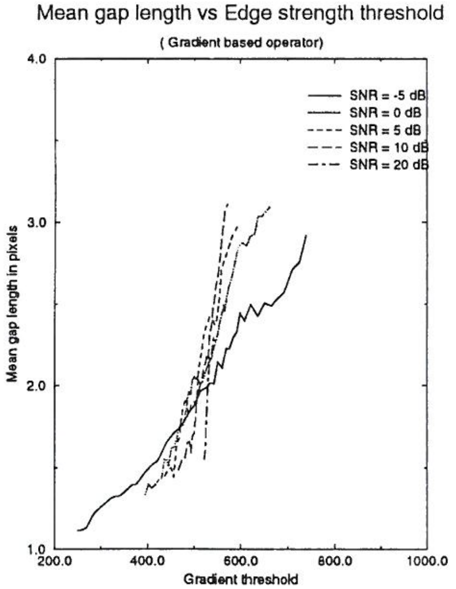

Figure 5: Plot of Mean gap run length vs Edge strength threshold for various signal to noise ratios. Orientation of the true edge was 15 degrees, Window size 5 by 5 for Gradient operator

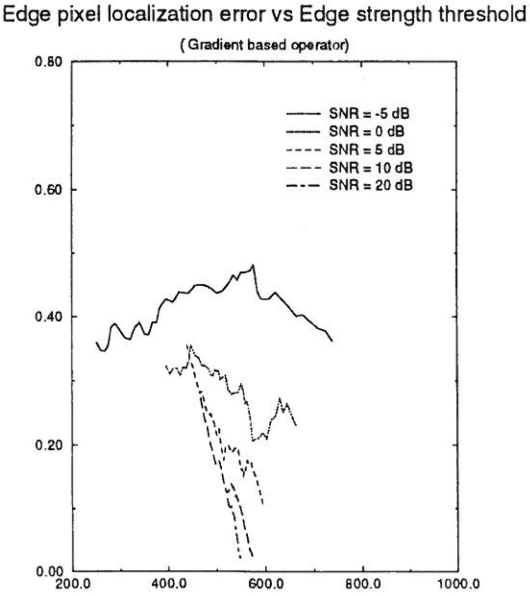

Figure 6: Plot of Mean pixel positional error vs Edge strength threshold for various signal to noise ratios. Orientation of the true edge was 15 degrees, Window size 5 by 5 for Gradient operator

7 Conclusion

In this paper we provided extensions of results provided in [1] and illustrated how one can relax the independence assumptions to derive random perturbation models that include the effects of correlation between neighboring gradient estimates. Under the assumption that the ideal data is corrupted with additive, independent additive gaussian noise, we derived expressions that describe the relationship between an edge gradient estimate at a given location and an edge gradient estimate for a neighboring pixel. We illustrated how the model line breakage process can be modeled as a Markov process whose parameters are functions of the true edge gradient, edge operator's neighborhood size, and the noise variance. Furthermore, we derived theoretical expressions for the mean positional error as a function of the neighborhood operator window size, noise variance, the width of the true ramp edge, and the true edge gradient. We also outlined an experimental protocol used for evaluating edge pixel positioning errors. The results from the experiments illustrate that gradient based edge schemes are superior (when edge localization is of interest) to the morphological scheme discussed in [3]. We also provided an algorithm to generate synthetic noisy edge images that utilize the statistics used in the experiments.

References

[1] V.Ramesh, R.M.Haralick, *"Random Perturbation Models and Performance Characterization in Computer Vision,"* Proceedings of the CVPR conference, June 1992.

[2] V.Ramesh, R.M.Haralick, *"Performance evaluation of edge operators,"* Special Session on Performance Evaluation of Modern Edge operators, Orlando Machine Vision and Robotics Conference, 20-24 April 1992.

[3] J.S.Lee, R.M.Haralick, and L.G.Shapiro, "Morphologic Edge Detection,"IEEE Journal of Robotics and Automation, Vol. RA-3, No.2, April 1987.

8 Appendix 1

We assume that the gray values in each neighborhood in the input image can be approximated by computing a planar fit. We assume further that each pixel in the input image is corrupted with additive Gaussian noise, $\eta(R, C)$, with zero mean and variance σ^2. Let $\hat{\alpha}_{R,C}$, $\hat{\beta}_{R,C}$, and $\hat{\gamma}_{R,C}$ denote the estimates for the coefficients best fitting plane that approximates the N by M neighborhood surrounding the center pixel specified by row and column coordinates (R, C). In this appendix we derive expressions describing the relationship between the estimates $\hat{\alpha}_{R+i,C+k}$ and $\hat{\alpha}_{R,C}$ for $\alpha_{R+i,C+k}$ and $\alpha_{R,C}$. Let α, β, and γ be the true plane coefficients. It can be easily shown that $\hat{\alpha}_{R,C}$ and $\hat{\beta}_{R,C}$ are equal to:

$$\hat{\alpha}_{R,C} = \alpha + \frac{\sum_{r=-N}^{N}\sum_{c=-M}^{M} r\eta(R,C)}{\sum_{r=-N}^{N}\sum_{c=-M}^{M} r^2} \quad (9)$$

$$\hat{\beta}_{R,C} = \beta + \frac{\sum_{r=-N}^{N}\sum_{c=-M}^{M} c\eta(R,C)}{\sum_{r=-N}^{N}\sum_{c=-M}^{M} c^2} \quad (10)$$

Now, we have: $\hat{\alpha}_{R+k,C+j} = \alpha +$:

$$\frac{\sum_{r=-N}^{N}\sum_{c=-M}^{M} r\eta(R+r+k, C+c+j)}{\sum_{r=-N}^{N}\sum_{c=-M}^{M} r^2} \quad (11)$$

The difference, $\hat{\alpha}_{R+k,C} - \hat{\alpha}_{R,C}$ is given by:

$$\begin{aligned} & \sum_{C'=C-M}^{C+M}\ \sum_{R'=R-N}^{R-N+k-1} (R-R')\eta(R',C') \quad (12) \\ + & \sum_{C'=C-M}^{C+M}\ \sum_{R'=R+N+1}^{R+N+k} \eta(R',C') \\ - & \sum_{C'=C-M}^{C+M}\ \sum_{R'=R-N+k}^{R+N} \eta(R',C') \end{aligned}$$

If we visualize the two overlapping neighborhoods, the first term in the above equation corresponds to the difference contributed by the nonoverlapping portion of the left mask, the second term corresponds to the contribution from the nonoverlapping portion of the right mask and the third term corresponds to the contribution from the overlapped portion of the two masks.

Similarly the difference between the $\hat{\beta}$'s, $\hat{\beta}_{R+k,C} - \hat{\beta}_{R,C}$, can be shown to be the ratio of:

$$\sum_{R'=R+N+1}^{R+N+K}\ \sum_{C'=C-M}^{C+M} c\eta(R',C') - \quad (13)$$

$$\sum_{R'=R-N}^{R-N+K+1}\ \sum_{C'=C-M}^{C+M} c\eta(R',C')$$

and $\sum_{r=-N}^{N}\sum_{c=-M}^{M} c^2$.

The above results can be summarized in a compact fashion using matrix notation. Let $\mathbf{V}_{\hat{\alpha}}$ denote a vector consisting of all the estimated $\hat{\alpha}$'s. Let A denote the matrix whose rows contain the values of the kernel used to estimate $\hat{\alpha}$. Let $\mathbf{E}_\eta$ denote the vector consisting of the additive Gaussian random variables $\eta(R, C)$'s. It can easily be seen that $\mathbf{V}_{\hat{\alpha}} = \mathbf{ME}_\eta + \alpha\mathbf{1}$ For example, the matrix A for $\begin{pmatrix} \hat{\alpha}_{R,C} \\ \hat{\alpha}_{R+1,C} \end{pmatrix}$ is given by the transpose of the following matrix:

$$\begin{pmatrix} -N & 0 \\ -N+1 & -N \\ -1 & -N+1 \\ \cdots & \cdots \\ 0 & 1 \\ 0 & -1 \\ 1 & 0 \\ \cdots & \cdots \\ N & N-1 \\ 0 & N \end{pmatrix} S \quad (14)$$

where $S = 1/\sum_{r=-N}^{N}\sum_{c=-M}^{M} r^2$, and the vector of η's are given by:

$$\begin{pmatrix} \eta_{R-N,C} \\ \cdots \\ \eta_{R+N+1,C} \end{pmatrix} \quad (15)$$

If the η's are Gaussian random variables with zero mean and covariance matrix Σ then we can see that the values in the vector $\mathbf{V}_{\hat{\alpha}}$ is distributed as a multivariate Gaussian random variable with mean $\alpha\mathbf{1}$ and covariance matrix $\mathbf{A\Sigma A'}$.

9 Appendix 2 – Procedure for generation of Noisy Edge images

The procedure for generation of noisy edge response images is explained below. First a ground truth edge image is created. This image is now perturbed to obtain the noisy edge image. Let,
G(r,c) denote the pixel values in the ground truth ideal edge image,
WSIZE denote the edge operator width

PMIS denote the probability of misdetection
Gap-mean denote the mean gap length
Gap-variance denote the gap length variance
Edge-run-mean denote the mean edge run length
Edge-run-variance denote the edge run length variance
Then:

```
for each pixel (R,C) in groundtruth image
   if ( G(R,C) == EDGELABEL )
    {
     /* Ground truth pixel is an
        edge pixel */

     X = Uniform(); /* Generate a uniform
                       random number between
                       0 and 1 */
     if ( X < Misdetect-Probability )
     {
       /* Edge pixel is to be perturbed
          according to the edge run and
          gap statistics */

       /* Currently, there are two modes of
          noisy edge generation. Mode 1
          corresponds to the generation of
          gaps and edge runs in the
          direction of edge gradient when
          the user provides the mean and
          variance values for the edge run
          lengths and gap lengths.
       */
       if ( MODE1 )
       {
        /* Generate a gap that is of size
           less the width of the edge
           operator.  The gap has to satisfy
           this constraint because the edge
           pixel is deemed to lie within the
           edge zone line (oriented along
           the edge gradient direction and
           is of width WSIZE pixels.) Gsample
           is a procedure that generates a
           sample from a Gaussian distribution.
        */

        AX[0] = Gsample(Gap-mean,
                        Gap-variance);
        while ( X > WSIZE )
          AX[0] = Gsample(Gap-mean,
                          Gap-variance);

        AY[0] = Gsample(Edge-run-mean,
                        Edge-run-variance);
        CURPOS = AX[0] + AY[0];
        if ( CURPOS > WSIZE )
         {
          /* Output data consists of a gap
             followed by a truncated edge run; */
         }
        else
         {
            i = 1;
            while ( CURPOS < WSIZE )
            {
              AX[i] = Gsample(Gap-mean,
                              Gap-variance);
              AY[i] = Gsample(Edge-run-mean,
                              Edge-run-variance);
              CURPOS += (AX[i] + AY[i]);
              i++;
            }
              /* Output data consists of
                 alternating gaps followed
                 by runs */
           }
       }

       /* MODE2 corresponds to the
          generation of an edge pixel run
          from the specification of the edge
          positional errors.
       */
       if ( MODE2)
       {
         X = Gsample(Edge-run-mean,
                     Edge-run-variance);
         Y = Gsample(pixel-error-mean,
                     pixel-error-variance);
         /* Output data now consists of a
            run of length X centered around
            (R,C) + (r',c'). The values r',c'
            are the coordinates of the pixel
            that is of distance Y from (R,C)
            along the line (oriented along
            the gradient direction).
         */
        }

     }
     else { /* Edge pixel was missed */ }
   }
  } /* End of "for each pixel
       in ground truth image" */
```