Automatic Selection of Tuning Parameters for Feature Extraction Sequences

Visvanathan Ramesh *
Robert M Haralick , Xining Zhang
Desika C Nadadur & Kenneth Thornton
Intelligent Systems Laboratory, Dept. of EE, FT-10
University of Washington, Seattle WA 98195
email:{rameshv,haralick,xzhang,desika,thornton}@george.ee.washington.edu

Abstract

Computer vision algorithms are composed of different sub-algorithms often applied in sequence. Previous work on performance characterization, [7], illustrated how random perturbation models, for the input and output data, can be setup at various stages of an algorithm sequence. In this paper we address the issue of how one could utilize these random perturbation models in order to automate the selection of free parameters used in an algorithm sequence. We consider an operation sequence that involves edge finding, linking, corner finding and matching. Appropriate prior distributions, for the parameters that describe the graytone/geometric characteristics of the image features, are specified and validated by using an annotation process. The annotation process involves the manual specification (outlining), for images in a training dataset, of the geometry and spatial extent of the image features. Statistics are gathered for parameters describing features of interest and non-interest (clutter features). The appropriate prior distributions are used to derive the theoretical expressions for feature detector performance over a given image population. These performance measures are then optimized to determine the tuning parameters for the feature detector(s).

Keywords: Performance Characterization, Feature (Edge/Corner) Extraction, Tuning Parameters, Low and Intermediate Level Vision

---

*Funding from ARPA is gratefully acknowledged. First author gratefully acknowledges part of his funding from IBM in the form of an IBM manufacturing Research Fellowship.

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, [7], theoretical models for the random perturbations in the output of a vision sequence, involving edge finding, edge linking and line fitting were presented. In other related work, [8], perturbation models, taking into account the correlation between neighboring pixels, were presented. Wang and Binford, [10], present theoretical characterization of edge operator performance and set the edge gradient threshold to be the threshold at which the false alarm rate is less than a specified value α. Hancock et al [2] present a scheme for adaptively selecting the hysteresis linking thresholds of Canny's, [1], hysteresis edge linking scheme. In contrast to the above two papers, our work, presented here, takes into account the prior distributions for parameters that describe the grayscale and geometric characteristics of image features.

In this paper we address the issue of how one could utilize random perturbation models, specified in [7] and [8], in order to automate the selection of free parameters used in an algorithm sequence. We consider an operation sequence that involves edge finding, linking, corner finding and matching. Appropriate prior distributions, for the parameters that describe the graytone/geometric characteristics of the image features, are specified and validated by using an annotation process, one in which a user outlines, for features of interest, the geometry and their spatial extent. Statistics are gathered for parameters describing clutter features that are not of interest. These statistics are used to compute the tuning parameters for the

feature detector(s) and the matching algorithm.

Specifically, our paper will illustrate a image analysis sequence involving edge detection, linking, and corner extraction. Selection of tuning parameters, for the algorithms employed for feature extraction, is achieved through an annotation process. Given a set of real images, the annotation process is used to label the groundtruth features. From the groundtruth we compute statistics, pertaining to gray level information as well as geometry, of the features of interest and features of non-interest. These statistics, combined with appropriate ideal and random peturbation models for image features, are used to set the tuning parameters of the feature extraction algorithms. This paper will not describe the protocol that is used for the generation of groundtruth, and the interested reader is referred to [13].

Given an algorithm sequence and a collection of images, in order to automate the selection of free parameters, we specify the following:

- Models for ideal image features at various steps.
- Perturbation models for these features.
- Prior distributions for image feature (grayscale/geometric) characteristics.

Given these models, and an appropriate criterion function, that measures the performance at the final stage of the algorithm sequence, we wish to select the free parameters that optimize the criterion function. This constitutes the optimization of the complete image understanding sequence. In this paper, we will describe how the optimization can be carried out for individual components ((e.g) edge detector, linker and corner detector) by specifying appropriate criterion functions at the outputs of each component.

This paper is organized as follows. We discuss the annotation process and outline the details of the algorithm sequence employed. Then, we provide a brief description of the perturbation models for data in various stages of the algorithm sequence. Further, we outline

models for the prior distributions of the interesting feature parameters and non-interest feature parameters. These distributions are used during the selection of free parameters for the algorithm sequence. Subsequent sections illustrate how these models are used for parameter selection.

2 Features and the annotation process

Recognition of 3D/2D objects takes place by recognizing features, i.e. feature detection, and their spatial relationships, i.e. matching. The class of features we consider here includes straight line segment features and dominant points (corners). In this section we discuss the nature of image features and the annotation scheme used.

These features have spatial extent and are characterized by two dimensions, gray levels and geometry. In order to detect features, some analysis of the gray scale values in a spatial neighborhood is required. Typically, algorithms for feature extraction make assumptions about the gray levels in spatial neighborhoods. For example, one assumption might be that the image graytone values are suitably approximated, locally, by a cubic polynomial. Another assumption could be that the intensity transition corresponding to a boundary may be modelled as a hyperbolic tan function. We make the measurements of the gray level structure for given features through an annotation process. This process will enable us to gather statistics about the gray level information as well as the geometry of features of interest (or of non-interest) to a specific application.

The annotation process is manual and a user interactively delineates features of interest and features of non-interest. In addition to the marking of a feature of interest, the annotation phase includes the delineation of the spatial neighborhood associated with the feature. This spatial neighborhood can be statistically analyzed to estimate quantities such

as: parameters of least squares fits to the gray scale values, over the entire neighborhood as well as on each side of the neighborhood, the residual fitting error for each kind of fit etc. Distributions of parameters that specify the geometry of the features can also be measured using the annotation process. For example, for line segment features, distributions of: center, length, orientation, starting point, and ending point, can be computed from the annotated groundtruth. From a suitably annotated set of training images it is, therefore, possible to measure, for each labeled feature, the spatial gray scale quantities, the geometric quantities, and quantities related to the feature detector, such as, edge or no edge. From these measurements we can estimate the distributions of these quantities.

3 Image Domain and Annotation

We utilize the model board image dataset provided as part of the ARPA RADIUS project to compute the statistics. The classes of 3D objects that are found in these images include: buildings, houses, roads, railroad tracks, trees, etc. Our model is a group of 2D line segments that are part of a building. We realize that this model is invalid on images where there are significant perspective effects and we wish to tackle the more general 3D problem at a later stage. We generated groundtruth for the model board imagery by using the protocol described in [13], and [12]. The objects of interest were the buildings. The groundtruth file, produced from the annotation process, gives us the outlines of the buildings. The outlines are labelled with id's corresponding to the id's in the model board reference diagram provided to us by ARPA. Figure 1 illustrates the original training images used with the annotations overlayed.

Using the groundtruth files and the original grayscale images we compute statistics that are used to set the tuning parameters for our algorithm sequence. We will first give a

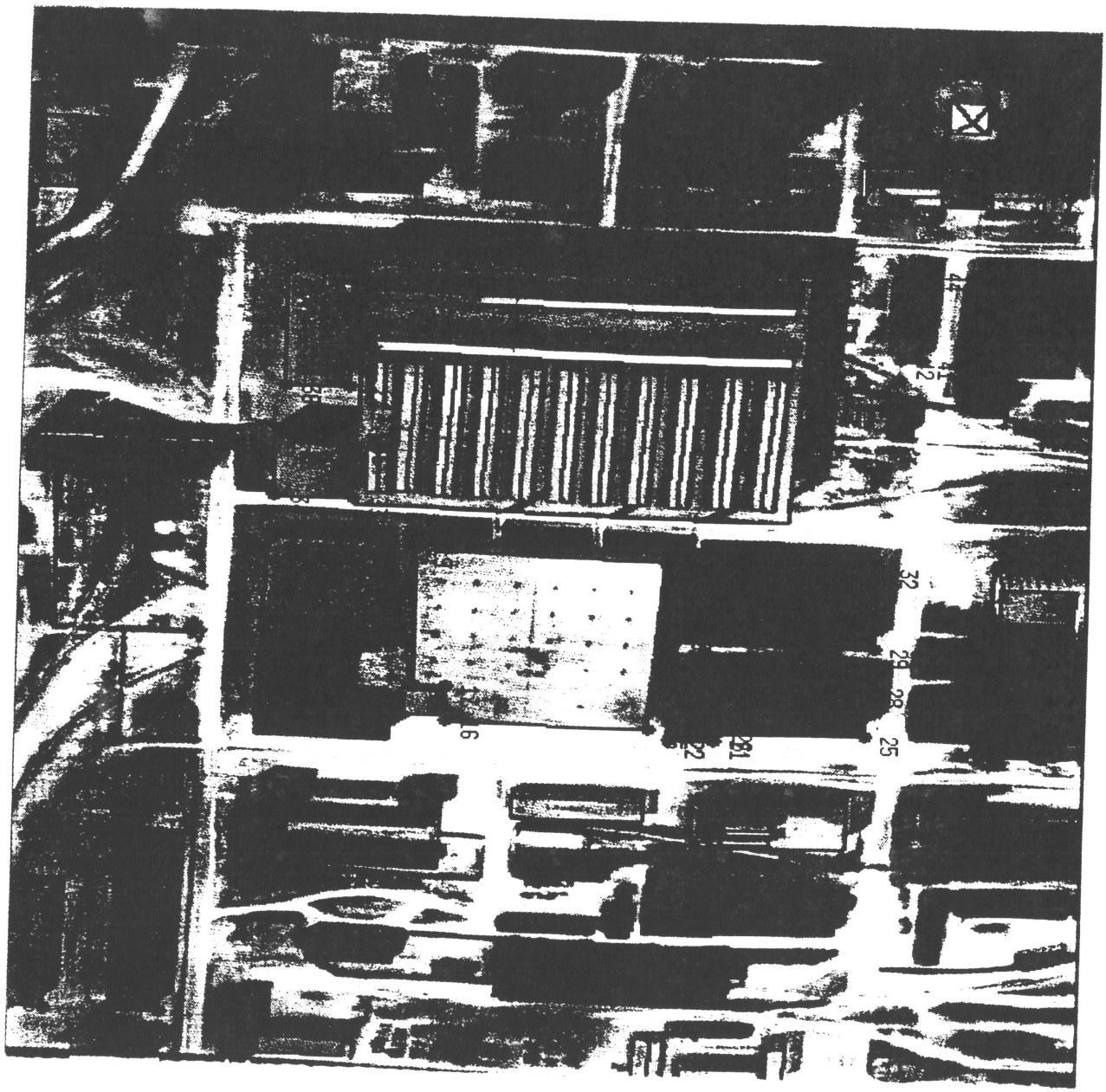


Figure 1: Annotations overlayed on original image

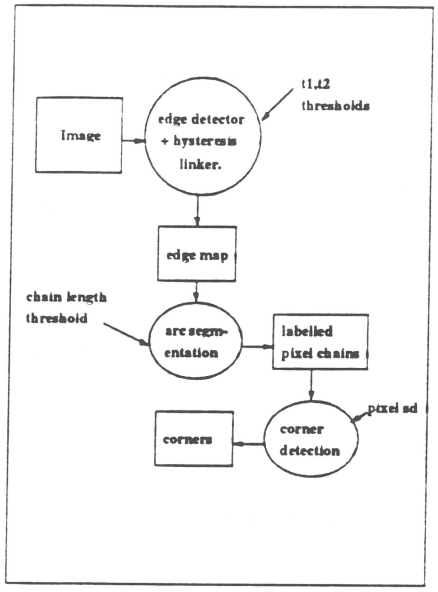


Figure 2: Image Analysis Algorithm Sequence

description of the image analysis algorithms employed. We will then outline the details of the statistics computed from the annotated and original grayscale images. Subsequent discussion will show how the statistics can be used to set the tuning parameters.

4 Algorithm Sequence

The algorithm sequence employed has the following components: an edge detector, a thinning algorithm, and a corner detector. Figure 4 illustrates the algorithm sequence used.

The edge detector that we employ is a two step procedure. In one step the gradient is estimated by using the slope facet operator of Haralick. The other step is Canny's hysteresis linking procedure that uses two thresholds. Canny, [1], uses two thresholds:

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

Specifically, starting from each potential edge candidate (pixels with gradient magnitude greater than T), the procedure performs a walk by examining each neighbor along the gradient direction and including them as edge pixels if the gradient magnitude is greater than T_2. The walk terminates when a pixel with gradient magnitude less than T_2 is encountered.

The tuning parameters employed in the edge detector are, therefore:

- T – Gradient threshold, T_2 – Second hysteresis threshold.
- K – Window size of the square window employed.

The thinning algorithm employed produces one pixel wide contours by recursive filtering of the edge output. For each pixel the procedure computes the connectivity number, using 4 or 8 neighborhood. The connectivity number is used to determine whether the pixel can be deleted without altering the image topology. The pixel value in the o/p image is assigned the background label if the pixel is deletable. Otherwise, the input pixel value is copied to the output. The neighborhood type, N (4/8 neighborhood) is the only tuning parameter used.

The output of the thinning step is passed to an arc segmentation procedure that identifies junctions and produces the component pixel chains. In order to filter out the pixel chains due to the clutter or due to the false positives of the edge operator, only pixel chains of length greater than a given threshold T_l are passed on to the corner detector.

The corner detector, utilized in the sequence, uses a bayesian scheme to incorporate prior information about corner angles. For specific details about the corner detector the interested reader is referred to [11]. Given an ordered sequence of points, the pixel chain, that belong to a digital arc, estimates of the break point, k, and the parameters of the line segments producing the corner are obtained by maximizing the aposteriori probability

$P(k, line - parameters| < pixelchain >)$. Zhang et al [11] models the true corner as being formed by the intersection of two ideal line segments. The perturbations on the detected edge pixels are assumed to be i.i.d. Gaussian samples, with zero mean and variance ${\sigma_e}^2$, along the direction of the edge intensity profile (In other words, the errors are along the direction normal to the line segment representing the edgel). This perturbation model is consistent with the edge location error model used in [8]. The prior distribution for the corner angle is assumed to be of the form of a Von Mises distribution with mean value $\theta = \pi/2$ and precision parameter κ. Since we are interested in finding rectangles or projections of rectangles, this distribution is appropriate for our problem domain. The prior distribution for the break point location along the given pixel chain is assumed to be a uniform distribution. The prior distributions for the true line parameters are assumed as follows. A, non-informative, uniform distribution in the interval $[0, 2\pi]$ is assumed for the true line orientation. Similarly, the distribution for the perpendicular distance of the line from the origin is assumed to be uniform within the image domain and zero elsewhere. Zhang et al,[11], show that the MAP estimate for the breakpoint location and the line parameters can be obtained as solutions to a system of nonlinear equations. The system is solved by using gradient descent. The solution provides the breakpoint and the parameters of the two lines forming the breakpoint. To handle the problem where multiple corners are present along the pixel chain, the algorithm is applied recursively to find breakpoints. In order to determine whether the breakpoint is significant or not, they compare the aposteriori probability estimate with a probability threshold T_p. The tuning parameters that are used in this algorithm are:

- Standard deviation σ_e, of the pixel error.
- Probability threshold T_p.

The output from the corner detector is a collection of points. This collection will then be

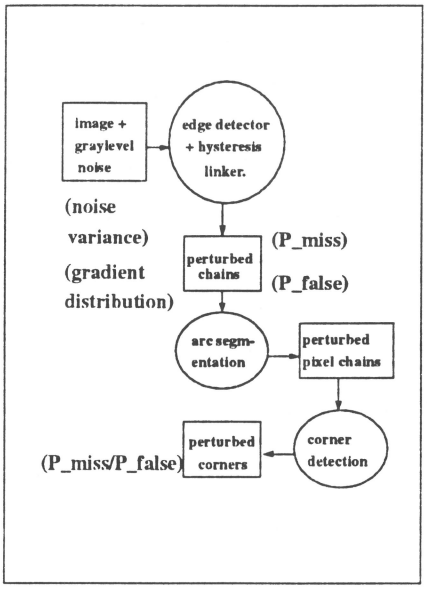


Figure 3: Perturbation models and Criterion Functions

matched to a collection of model points. We do not discuss the matching algorithm in this paper.

5 Perturbation Models

In this section we provide brief description of the perturbation models for data at each stage of the algorithm sequence. Figure 5 illustrates the various perturbation models and the criterion functions at each stage of the algorithm sequence.

5.1 Perturbation models for I/P at the edge detector

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. In [7] we used a non-edge idealization that

is a flat graytone surface and assumed a random perturbation model where each pixel value was 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. This model is appropriate when we are interested in evaluating edge detectors.

In reality, there are intensity transitions in images arising from structures that are not of interest to the matching procedure. Boundaries arising from these structures are the clutter segments that are present in the input to a matching algorithm. We assume that, for clutter features, the squared edge gradient has an exponential distribution. We assume that the edge gradient for features of interest, has a Gamma distribution.

5.2 Perturbation models for O/P at the edge detector and linker

Our boundary idealization is that of a one pixel wide pixel chain. The process that randomly perturbs this ideal boundary is modelled as a renewal process with alternating segment and gap intervals. The interval lengths, under certain independence assumptions, were shown to be distributed as truncated exponential distributions [7]. The distributions have this form only if the values of the ideal edge gradient along a given boundary are the same. This assumption can be relaxed to include boundaries where shading effects are present. The distribution of the intervals will then depend on the the gradient profile along the boundary. In [8], it was shown that one can incorporate dependencies between neighboring pixel edge gradient estimates by a Markov chain of a given order.

We gave a brief description of the boundary fragmentation process above and did not describe positional errors. We assume that the edge pixel position error, measured along the gradient direction, can be approximated by a Gaussian distribution with zero mean and variance $\sigma_e{}^2$.

The output data consists of perturbed, perhaps fragmented, interesting and clutter fea-

tures (pixel chains). The segment length distribution, gap length distribution, the edge pixel error distribution, and the orientation error distribution characterize the errors at this stage.

5.3 Perturbation models for O/P and I/P data of the Corner Detector

The input to the corner detector is a collection of perturbed pixelchains belonging to features of interest and clutter features. The perturbation on each individual pixel of the pixel chain is described by the edge pixel error distribution described in the previous section. The output from the corner detector is a collection of feature points (belonging to the interesting as well as non-interesting objects). The exact nature of the probability distribution of the corner location errors is the subject of another paper [9]. The standard deviation in the corner pixel location is described by a parameter σ_c. In order to solve the model matching problem, we have to specify the probability distributions for the spatial location of the model points generated for clutter and objects of interest. We do not discuss them in this paper.

6 Statistics Computation from Groundtruth and Parameter Selection

In this section we will provide details of the statistics that are computed from the groundtruth data. These statistics are used to specify the tuning parameters of the algorithms in the sequence. We organize this section into subsections identifying the statistics required for setting the parameters for each algorithm in the sequence. Figure 6 illustrates the methods used for computing the statistics.

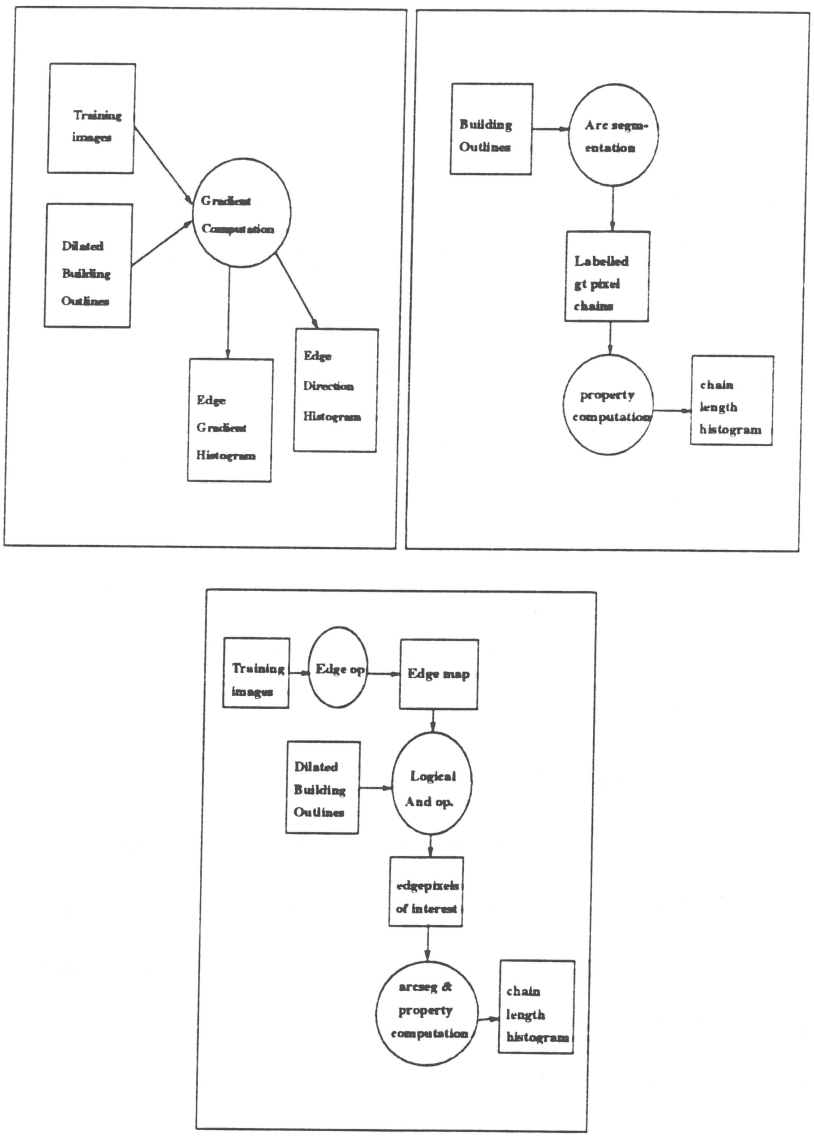


Figure 4: Statistics computation from Training Data. Top left figure illustrates statistics related to edges. Top right figure illustrates chain length statistics computation from groundtruth chains. Bottom figure illustrates chain length statistics computation for clutter features.

6.1 Edge Detection – Statistics and Threshold Selection

We compute the following statistics in order to set the parameters for the edge detector:

- The assumption in [7] was that the ideal image was corrupted with i.i.d. zero mean Gaussian samples having noise variance σ^2. This parameter has to be estimated from the annotations. Let the annotated edge image be I_E. We perform a morphological dilation on I_E by a 5 by 5 box structuring element. The resulting image is used as a mask. We randomly choose 5 by 5 neighborhoods, on the image, that do not overlap with any pixel on the mask and compute the least squares planar fit over the neighborhoods. Let the average residual error be e_{avg}. Then the estimate $\hat{\sigma}^2$ is given by: $e_{avg}/22$. Here 22 is the number of degrees of freedom.

- Our threshold selection scheme assumes that the squared edge gradient distribution for features of non-interest is an exponential distribution with parameter λ_g. In addition, the method assumes that the squared edge gradient distribution for edge features of interest can be modelled as a Gamma distribution with parameters a_g and b_g. These parameters have to be estimated from the annotated data. We obtain the estimate $\hat{\lambda}_g$ by computing the average squared gradient obtained from the least squares planar fitting done over the randomly chosen 5 by 5 neighborhoods mentioned above. Estimates of parameters of the Gamma distribution, $\hat{a}_g$ and $\hat{b}_g$, are obtained by using the method of moments [5]. By computing the least square fits over 5 by 5 neighborhoods centered on the locations specified by edge pixels in the annotated images we obtain estimates of the squared gradient values. Denoting these samples as: $g_1, g_2, g_3, \ldots, g_N$, where N is the number of edge pixels in the annotated image, the second and third

moments, denoted by m_2 and m_3, are computed. Then:

$$\hat{a}_g = \frac{4{m_2}^3}{{m_3}^2} \quad (1)$$

$$\hat{b}_g = \frac{m_3}{2m_2} \quad (2)$$

It is also possible to obtain maximum likelihood estimates for a_g and b_g.

- In order to select the appropriate neighborhood size for the edge operator we need to compute the distribution of the true edge scale from the annotated data set. We, instead, use the nearest neighbor distance distribution between edge features of interest. We define the term "nearest neighbor edge pixel" for a groundtruth edge pixel as the closest edge pixel along the direction of the edge intensity profile. The distance distribution between nearest neighbors specify the separation between adjacent edges. The edge operator window size that is used is equal to the minimum nearest neighbor distance that is computed. We used a 5 by 5 neighborhood for the RADIUS data. We are in the process of justifying the selection of this parameter by using the empirical distribution of nearest neighbor edge pixel distance.

The gradient threshold is automatically selected by using the theoretical derivation given below.

6.1.1 Threshold Parameter Selection for Edge operator

In this section we address the problem of automatic selection of the parameters for the edge operator in the algorithm sequence used for the demonstration. In order to derive the appropriate threshold we will utilize the theoretical expressions derived in [7]. It was shown in [7] that under the assumption that the image pixels were corrupted with additive i.i.d Gaussian noise having zero mean and variance σ^2, the distribution of the gradient estimate

is a non-central chi-squared distribution. Specifically, for a square neighborhood, it was shown that $\hat{G}^2 \sum_{(r,c)} r^2/\sigma^2$ is a non-central chisquared distribution with 2 degrees of freedom and non-centrality parameter $C = g^2 \sum_{(r,c)} r^2/\sigma^2$. Here g is the true gradient value in the neighborhood. Let ${\sigma_1}^2 = \sum_{(r,c)} r^2/\sigma^2$ and ${\hat{G}'}^2 = \hat{G}^2/{\sigma_1}^2$. Then $C = g^2/{\sigma_1}^2$.

The probability density function for a non-central chisquare distribution with number of degrees of freedom v and non-centrality parameter λ is given by:

$$p(x) = \frac{1}{2}(x/\lambda)^{\frac{1}{4}(v-2)} I_{\frac{1}{2}(v-2)}(\sqrt{\lambda x}) e^{[-\frac{1}{2}(\lambda+x)]} \ (x > 0) \tag{3}$$

where I_k denotes the modified Bessel function of the first kind and order k.

Taking account of the fact that $v = 2$ we can write the probability the density function $P({\hat{G}'}^2 = x^2|g, \sigma)$ as:

$$P({\hat{G}'}^2 = x^2|g, \sigma_1) = \frac{1}{2} I_0(\frac{gx}{\sigma_1}) e^{-\frac{1}{2}\left(\frac{g^2}{{\sigma_1}^2}+x^2\right)} \tag{4}$$

In reality, there is a distribution on the observed values of g and σ. For computing the gradient threshold we specify a distribution for the edge gradient for intensity edges of non-interest. We assume that the squared edge gradient distribution can be modelled as a exponential distribution with mean parameter λ_g. Under this assumption the joint pdf for ${\hat{G}'}^2$ and G_t, the random variable describing the true gradient, is given by:

$$P({\hat{G}'}^2 = x^2, G_t = g|\sigma_1) = \frac{1}{2\lambda_g} I_0\left(\frac{gx}{\sigma_1}\right) e^{-\frac{1}{2}\left(\frac{g^2}{{\sigma_1}^2}+x^2\right)} e^{-\frac{g^2}{\lambda_g}} \tag{5}$$

In order to obtain the conditional density of ${\hat{G}'}^2/\sigma_1^2$, given the σ_1, we need to integrate the above equation with respect to g^2. Making use of the fact that $I_v(w)$ is given by the equation:

$$I_v(w) = \sum_{m=0}^{\infty} \frac{1}{m!\Gamma(m+v+1)} \left(\frac{w}{2}\right)^{2m+v} \tag{6}$$

the right hand side of the above equation becomes:

$$\frac{1}{2\lambda_g}\int I_0\left(\frac{gx}{\sigma_1}\right)e^{-\frac{1}{2}\left(\frac{g^2}{{\sigma_1}^2}+x^2\right)}e^{-\frac{g^2}{\lambda_g}}dg^2 \tag{7}$$

$$= \frac{1}{2\lambda_g}e^{-(x^2/2)}\int\sum_{m=0}^{\infty}\frac{1}{m!\Gamma(m+1)}\left(\frac{gx}{2\sigma_1}\right)^{2m}e^{-\frac{g^2}{2}\left(\frac{1}{{\sigma_1}^2}+\frac{2}{\lambda_g}\right)}dg^2$$

Interchanging the order of the integral and the summation, and taking the g^2 terms together we get:

$$\frac{1}{2\lambda_g}e^{-(x^2/2)}\sum_{m=0}^{\infty}\frac{1}{m!\Gamma(m+1)}\left(\frac{x}{2\sigma_1}\right)^{2m}\int g^{2m}e^{-\frac{g^2}{2}\left(\frac{1}{{\sigma_1}^2}+\frac{2}{\lambda_g}\right)}dg^2 \tag{8}$$

It can be easily shown, by making use of the fact that $\Gamma(k)=\int z^{k-1}e^{-z}dz$, that the integral in the above expression evaluates to:

$$\frac{\Gamma(m+1)}{(C(\sigma_1,\lambda_g)^{m+1})} \tag{9}$$

where: $C(\sigma_1,\lambda_g)$ is given by:

$$\frac{1}{2{\sigma_1}^2}+\frac{1}{\lambda_g}. \tag{10}$$

This implies that we can write the conditional density as:

$$P(\hat{G'}^2=x^2|\sigma_1)=\frac{1}{2\lambda_g C(\sigma_1,\lambda_g)}e^{-(x^2/2)}\sum_{m=0}^{\infty}\frac{1}{m!}\left(\frac{x^2}{4{\sigma_1}^2C(\sigma_1,\lambda_g)}\right)^m \tag{11}$$

Making note of the fact that the summation is actually the expanded form of an exponential function, and simplifying, we get:

$$P(\hat{G'}^2=x^2|\sigma_1)=\frac{1}{2\lambda_g C(\sigma_1,\lambda_g)}e^{-\frac{x^2}{2\lambda_g C(\sigma_1,\lambda_g)}} \tag{12}$$

It can be seen from equation 12 that the conditional density is an exponential density function. In order to compute the gradient threshold, t, corresponding to a particular probability of false alarm, α, we just set:

$$\int_t^\infty \frac{1}{2\lambda_g C(\sigma_1, \lambda_g)} e^{-\frac{z}{2\lambda_g C(\sigma_1, \lambda_g)}} dz \leq \alpha \tag{13}$$

This corresponds to setting:

$$e^{-\left(\frac{t}{2\lambda_g C(\sigma_1, \lambda_g)}\right)} = \alpha \tag{14}$$

. The value of t is therefore: $2C(\sigma_1, \lambda_g)\lambda_g(-log\alpha)$. Since $t = T^2/{\sigma_1}^2$ corresponds to the parameter for the scaled variable ${\hat{G'}}^2$, the appropriate threshold T^2 corresponding to $\hat{G}^2$ is given by:

$$T^2 = \left(\lambda_g + \frac{2\sigma^2}{\sum_{(r,c)} r^2}\right)(-log\alpha) \tag{15}$$

The threshold parameter can be seen as a product of one factor that is dependent on the probability of false alarm and another that is dependent on the noise variance and the non-edge gradient distribution. The term $\sum_{(r,c)} r^2$ depends on the neighborhood size employed in the operator. The threshold necessary, for the given false alarm, will be lower, when large neighborhood sizes are employed. Intuitively, we can afford to lower the threshold since a large window size implies that there is considerable reduction in the noise variance.

Discussion

The above equations were derived assuming a single noise variance value. One can assume a prior distribution for the noise variance and compute the marginal density for ${\hat{G'}}^2$. We can then derive the optimum threshold required for a given false alarm rate. We did not pay much attention to the misdetect rate here. In reality, one may wish to find the threshold that minimizes a convex combination of the probability of false alarm and the probability

of misdetection. In the above section the expression for the threshold was derived assuming a prior distribution of non-edge gradient values. One can pose the question differently, and try to determine thresholds adaptively over the entire image. If our exponential model, for the prior gradient distribution of non-interesting features, is not valid, one can following a similar derivation to incorporate the alternative model. The derivation in the next section illustrates the situation when the prior gradient distribution is a Gamma distribution. If the prior distribution is a Gamma distribution, we can use the derivation in the next section to compute the probability of false alarm and set the appropriate threshold. The following section will illustrate how we can set the hysteresis threshold of Canny's hysteresis linking procedure by minimizing the probability of misdetection of an edge operator.

6.1.2 Probability of Misdetection for a specified threshold t

In this section we derive the expression for the probability of misdetection of an edge operator over an image population. Specifically, we determine the probability of misdetection at a given threshold t by assuming a prior distribution for the edge gradient for features of interest. A look at the RADIUS imagery reveals that the edge gradients for features (building boundaries) of interest vary from low values, in the regions where shadow effects are present, to high values, in the boundary between roof tops and vertical planes of the building. Hence we model the prior distribution of the squared edge gradient as a two parameter Gamma distribution with parameters a_g and b_g. The general Gamma distribution has three parameters. We set the third parameter describing the left most cutoff point of the distribution to be zero, since we assume that there are features of interest that have significantly low gradient values.

That is:

$$P(g^2) = \frac{(g^2)^{(a_g-1)} e^{\frac{-g^2}{b_g}}}{b_g^{a_g}\Gamma(a_g)} \tag{16}$$

Under this assumption, the joint density of $\hat{G'}^2$ and G_t^2 (the r.v. corresponding to the true gradient value), given σ_1 can be written as:

$$P(\hat{G'}^2 = x^2, G_t^2 = g^2) = \frac{e^{-\frac{x^2}{2}}}{2{b_g}^{a_g}\Gamma(a_g)} I_0\left(\frac{gx}{\sigma_1}\right) e^{-\left(g^2\left(\frac{1}{2{\sigma_1}^2}+\frac{1}{b_g}\right)\right)} (g^2)^{(a_g-1)} \tag{17}$$

Integrating the above equation with respect to g^2 results in the expression:

$$\frac{e^{-\frac{x^2}{2}}}{2{b_g}^{a_g}\Gamma(a_g)} \sum_{m=0}^{\infty} \left(\frac{x^{2m}}{2^{2m}m!\Gamma(m+1)}\right) \frac{\Gamma(m+a_g)}{C(\sigma_1, b_g)^{m+a_g}} \tag{18}$$

where: $C(\sigma_1, b_g) = \frac{1}{2{\sigma_1}^2} + \frac{1}{b_g}$. As we can see, this expression is not in a nice closed form. The probability of misdetection for a given threshold t^2 can however be expressed in a computable form. The probability of misdetection is given by the integral of the above expression between limits 0 and t^2. Hence grouping all the terms involving x^2, integrating and simplifying gives:

$$P_{misdetection} = \frac{1}{{b_g}^{a_g}\Gamma(a_g)} \sum_{m=0}^{\infty} \frac{\Gamma(m+a_g) I(t^2, m+1)}{(2\sigma_1)^m m! C(\sigma_1, b_g)^{m+a_g}\Gamma(m+1)} \tag{19}$$

Here, $I(x,k)$ is the incomplete Gamma integral given by the equation:

$$I(x,k) = \int_0^x z^{k-1} e^{-z} dz \tag{20}$$

Thus, knowing the threshold, t^2, the parameters of the prior distribution, a_g, b_g, the noise variance σ^2, and the neighborhood size of the window, we can compute the probability of misdetection.

6.1.3 Selection of second Hysteresis threshold T_2

In this section we address the problem of selection of the hysteresis threshold employed. For a chosen T, we showed, in the section above, that one could compute the probability of misdetection. In order to select the second hysteresis threshold one has to use the information about the standard deviation in the gradient along the true edge pixels. It is clear that the lower bound for the probability of misdetection, after hysteresis linking, is given by the probability of misdetection for a threshold of T_2. It was shown in [6] that, assuming independence between pixel gradient estimates, the probability of misdetection after the hysteresis linking step is given by the expression:

$$F_G(T_2) + (F_G(T) - F_G(T_2))F_G(T)^{W-1} \tag{21}$$

where $F_G(t)$ represents the cumulative distribution of the gradient magnitude, and W is the width used for hysteresis linking. We select the hysteresis threshold by making the $P_{misdetection}(T_2) = \alpha_2$. Note that the second term in the above expression is negligible when the probability of misdetection at threshold T, $F_G(T)$, is small. In this case, the probability of misdetection after the linking is given by $F_G(T_2)$. We choose T_2 as the threshold that sets $P_{misdetection}$ to be less a small number than α_2. That is we set:

$$\alpha_2 = \frac{1}{{b_g}^{a_g}\Gamma(a_g)} \sum_{m=0}^{\infty} \frac{\Gamma(m + a_g)I({T_2}^2, m + 1)}{(2\sigma_1)^m m! C(\sigma_1, b_g)^{m+a_g}\Gamma(m + 1)} \tag{22}$$

to determine T_2.

In the special case, where the distribution for the squared edge gradient is also exponential

with parameter λ_e, the threshold ${T_2}^2$ can be shown to be equal to:

$$min\left(T^2, \left(\lambda_e + \frac{2\sigma^2}{\sum_{(r,c)} r^2}\right)(-log(1-\alpha_2))\right) \tag{23}$$

6.1.4 Estimate of edge pixel localization error

We obtain an estimate of the edge pixel localization error from training data by following the specific protocol described in [8]. Specifically, for the annotatated data set, we apply the edge detector, with the thresholds derived as above, and compute the mean deviation of the edge pixel location. This estimate $\hat{\sigma}_e$ is used as input to the corner finder.

6.2 Thinning – Parameter Selection

The thinning algorithm requires one parameter that specifies whether eight or four connected neighbors are used, in order to compute the connectivity number, in the algorithm. We use eight connected neighbors.

6.3 Chain Length Threshold selection

Before we set the chain length threshold, we process the groundtruth images to obtain the empirical distributions for the lengths of chains of interest. To determine the empirical distribution for lengths of non-interesting pixel chains we set the edge gradient threshold at a value corresponding to $\lambda_g = 0$ and set the false alarm rate, α, to 0.2. in equation 24. This threshold while taking into account the noise variance, treats all features as interesting features. That is:

$$T^2 = \left(\frac{2\sigma^2}{\sum_{(r,c)} r^2}\right)(-log\alpha) \tag{24}$$

We use this edge gradient threshold T and the hysteresis threshold of T_2 corresponding to a misdetect rate of 0.01, to obtain edge images. We dilate the annotated images by a 5 by 5 structuring element and invert the result to produce an image that serves as a mask for areas of non-interest. We then obtain the empirical distribution for the chains in the areas of non-interest. Figure 5 illustrates an example of the image that is used to compute statistics of pixel-chains. This image was obtained as the result of processing the input image with a edge threshold parameter for a false alarm rate of 20 percent, thus generating clutter chains as well as chains of interest. Note that part of the pixel chains belonging to the roof of a building are missing because of error in the outlining process. This error is of the order of approximately 3 pixels.

The chain length threshold is set at a value such that the sum of probability of missing features of interest and the probability of introducing clutter features is minimized. Suppose the chain length distribution for features of interest is an exponential distributions with mean parameter λ_1 and offset l_1. Let the chain length distribution for features of non-interest be an exponential distribution with mean parameter λ_2 and offset 0. We can easily show that the threshold that minimizes the sum of the probability of false alarm and probability of misdetection is given by:

$$T_l = \frac{log(1 - e^{-\lambda_1 l_1}) + log(\lambda_2/\lambda_1) - \lambda_1 l_1}{\lambda_2 - \lambda_1} \tag{25}$$

6.4 Corner Extraction – Statistics & Parameter Selection

We have seen that the Bayesian corner detector uses prior distributions for the line parameters such as perpendicular distance and orientation. The detector assumes a Von Mises distribution for the corner angle. These assumptions were verified by computing the empirical distributions from the annotated data. We have seen, in a previous section, how the

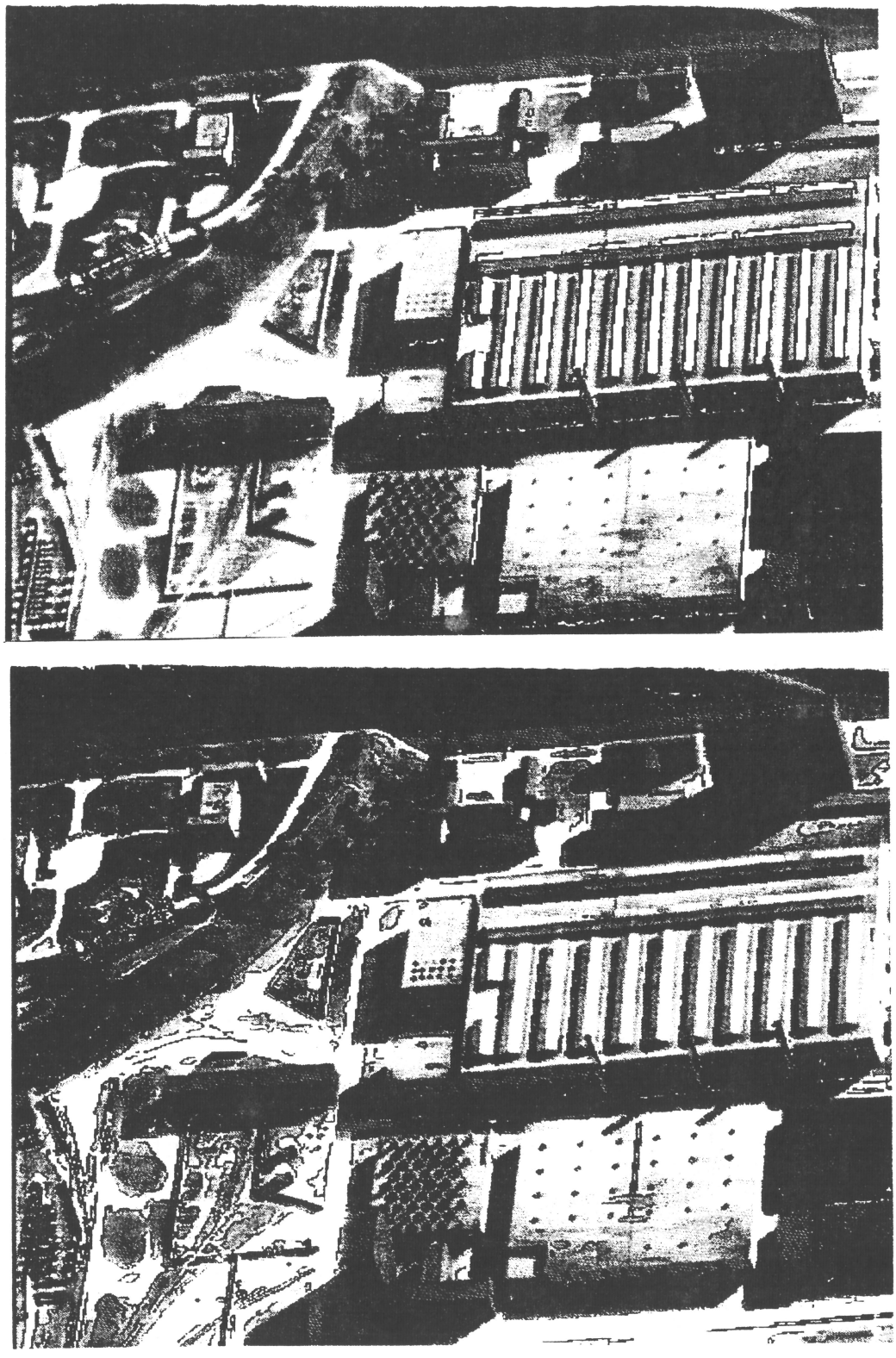


Figure 5: Image illustrating detected chains of interest (top) and non-interest (bottom))

parameter for the pixel error is obtained from the edge operator output.

Assuming that the input to the corner detector is a perturbed version of a ideal line segment, we can derive the probability of false alarm for a given probability threshold T_p. Let N denote the number of pixels in the observed pixel chain. Let the logarithm of the prior probability of the included corner angle be denoted by $g(\theta)$. Let K denote the logarithm of the normalizing constant that is part of the objective function that is minimized. Then, it is shown in [9] that the probability of false alarm of the corner detector is given by the probability:

$$Prob(\hat{X} > \frac{(-logT_p - g(0) + K)}{{\sigma_e}^2}) \tag{26}$$

where $\hat{X}$ is a chisquare random variable with N degrees of freedom and σ_e is the standard deviation of the edge pixel location error. The probability threshold T_p that sets the probability of false alarm to be below a given level α is:

$$T_p = e^{K-g(0)} e^{\left(({\chi_I}^2)^{-1}(\alpha){\sigma_e}^2\right)} \tag{27}$$

where $(\chi_I^2)^{-1}(\alpha)$ gives the value x that makes the tail probability equal to α. Since K is a function of σ_e the edge pixel error standard deviation, the optimum threshold for the corner detector is a function of σ_e and the number of points on a given pixel chain. For each corner point extracted, the variance of the estimated location can be computed. The error propagation for this is given in another paper by Zhang et al [14].

6.5 Results on RADIUS Dataset

In this section we provide some of the results obtained by following the above methodology on RADIUS imagery. Following the methodology we estimated the edge gradient distribution(s), the noise variance, and computed the threshold for various false alarm rates ((e.g)

20, 10, 5, 1 percent).

6.5.1 Empirical Distributions for Features

The empirical distributions of the edge gradient for features of interest and features of non-interest were obtained from the annotated images. Figure 6 shows the distributions. For areas of non-interest, there is a peak in the empirical distribution around edge gradient of zero and the distribution is approximated by an exponential distribution. This approximation is not exactly correct, because the empirical distribution has a fat tail (since there were few regions in areas of non-interest that have significantly high edge gradient). A more appropriate distribution is a mixture distribution, a mixture of exponential and linear. We are in the process of deriving the appropriate expressions for the thresholds under this model. It can be seen that the empirical distribution for the edge gradient in areas of interest is a fat-tailed distribution. It was seen that the Gamma distribution approximates the empirical distribution poorly for low values of edge gradient, and the approximation is good at intermediate values of edge gradient and at the tail. We are in the process of obtaining theoretical expressions for the probability of false alarm when the prior distribution is a mixture distribution.

It can be seen from figure 7 that the chain length distributions are approximated as exponential distributions. The approximation is appropriate for clutter chain lengths and is less appropriate for chains of interest. Again, it is possible to model the chain length distribution for features of interest as a mixture distribution. We are in the process of doing this. We will then validate our distribution assumptions by performing a Kolmogorov-Smirnov test.

The results obtained for various false alarm rates are shown in figure 8. Note that part of the segments belonging to the building are missing. We see that most of the segments are

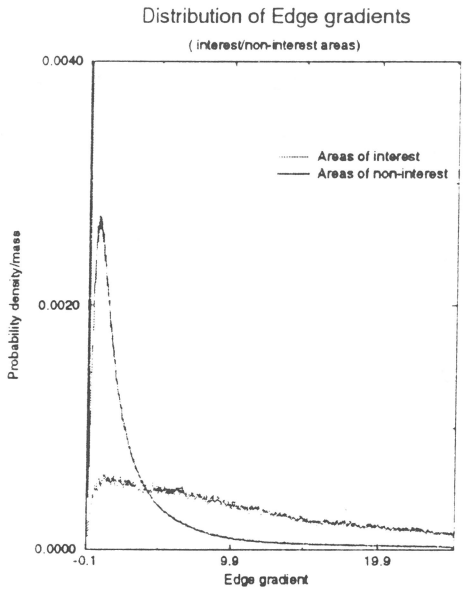
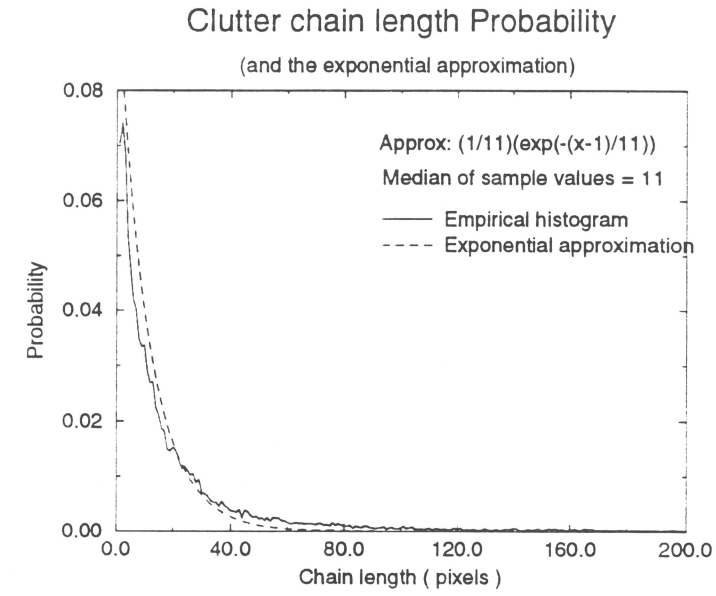


Figure 6: Empirical Distribution of Edge Gradients

detected when the mean gradient (for clutter) is set to zero (thus setting the threshold only based on noise variance). On the other hand, when we take into account that the clutter gradient distribution, the threshold for a given false alarm rate is significantly higher and more segments that are part of the building are missed. This is because, the thresholds have to be significantly higher in order to reduce the amount of clutter pixels (as the mean of the prior distribution of the edge gradient of clutter features is quite high). High level vision algorithms can obtain a minimal set of features to generate matching hypotheses by setting the false alarm rate to a low value (say 5 percent). During the hypothesis verification stage, the false alarm rate can be set at a higher value (say 15 percent) in order to detect more features around regions of matches.

The example image shown in this paper is just one representative result that is obtained. In order to verify that our algorithm produces optimal results over an entire dataset, we should use the performance on the entire population of the images as an indicator of how well our scheme works. We are in the process of obtaining performance curves, for the RA-



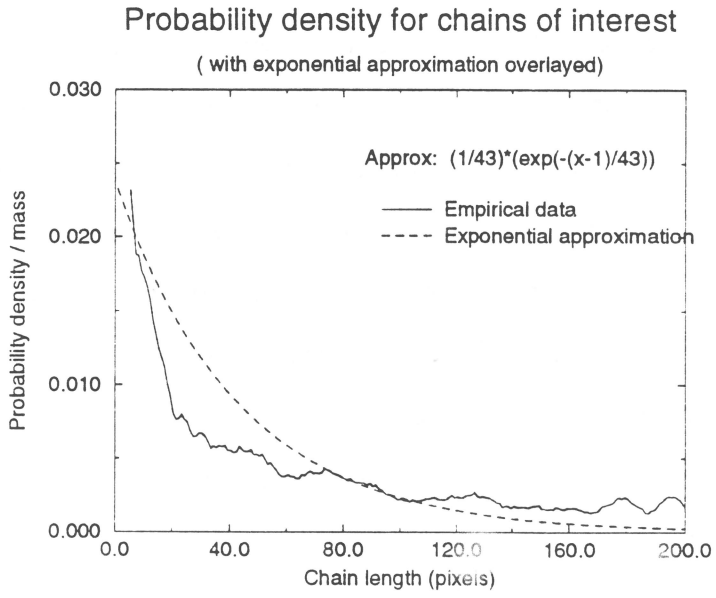


Figure 7: Empirical Distribution of Chain lengths

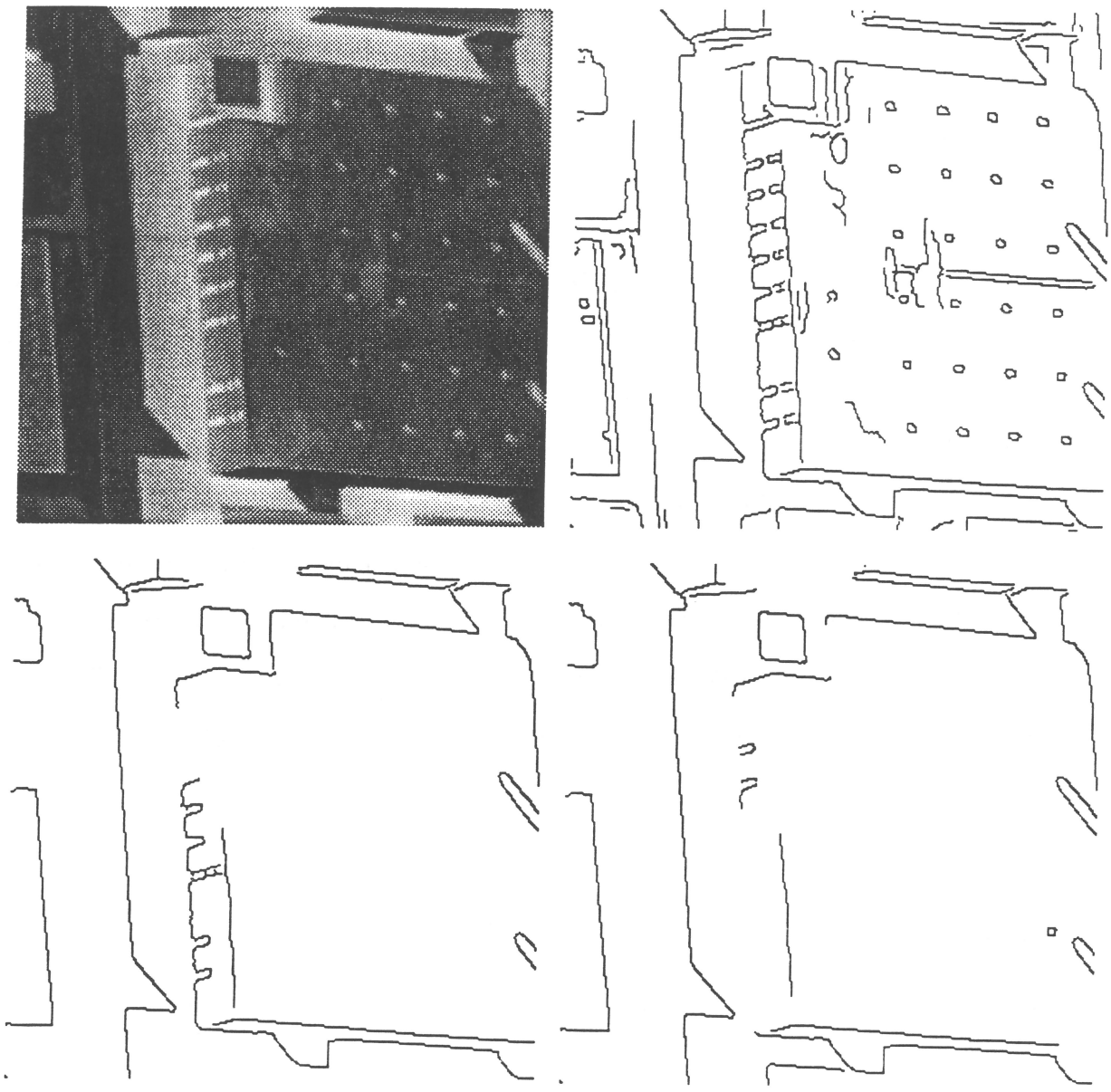


Figure 8: Pixel chains detected for an example image. Subimage of a model board image (top left), Detected edges by setting $\lambda_g = 0$ and false alarm rate 10 percent (top right), Detected edges for false alarm rate of 10 percent (bottom left), Detected edges for false alarm rate of 5 percent (bottom right).

DIUS data set, comparing the estimated (feature attributes) quantities and the groundtruth quantities.

7 Conclusion

In this paper we illustrated how one can automatically select the parameters for a feature extraction sequence involving edge finding, linking, and corner detection. We stated how we annotate images and use the annotations to estimate the distributions of parameters that describe the features. We then illustrated how the tuning parameters for the algorithms were set using the annotations. We are currently working on automatically setting the parameters for the matching algorithm. Since we have the groundtruth for the RADIUS imagery, we can also provide curves describing algorithm performance on the real dataset. In this paper, each step in the algorithm sequence was individually optimized. However, it is well known that, optimization of each individual component a system does not necessarily give us the optimal result for the complete system. We are in the process of combining the results in order to do the total optimization problem.

References

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

[2] E. R. Hancock and J. Kittler, "Adaptive Estimation of Hysteresis Thresholds ," Proceedings of the 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 196-201.

[3] R.M.Haralick, "Digital Step Edges from Zero Crossings of Second Derivative," IEEE Transactions on PAMI, 1984.

[4] R.M.Haralick and X. Zhang, "Error propagation for Computer Vision," Document in preparation.

[5] N.L.Johnson and S.Kotz, *Continuous Univariate Distributions*, 1970, John Wiley and Sons, NY.

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

[7] V. Ramesh and R.M.Haralick, "Random Perturbation Models & Performance Characterization in Computer Vision," Proceedings of the CVPR conference held at Champaign, IL, June 92, pp. 521-527.

[8] V. Ramesh and R.M.Haralick, "Performance Evaluation of Edge Operators," Proceedings of the DARPA IUW 93 held at Wash DC, April 93, pp. 1071-79.

[9] V. Ramesh, X. Zhang and R.M.Haralick, "Theoretical Performance Analysis of a Corner Detection Scheme," Document in preparation.

[10] S. Wang and T. Binford, "Local Step Edge Estimation – A New Algorithm, Statistical Model and Performance Evaluation," Proceedings of the ARPA IU Workshop, Wash DC, April 1993, pp.1063-70.

[11] X.Zhang, "Bayesian Corner Detection," Intelligent Systems Lab Internal Report, Aug 93.

[12] X. Zhang, "Model Board Imagery Groundtruth Generation – User manual," Intelligent Systems Lab Internal Report, Aug 93.

[13] X. Zhang and V. Ramesh, "Model Board Imagery Groundtruth Generation Protocol," Intelligent Systems Lab Internal Report, Department of EE, University of Washington, Seattle, WA. Aug 93.

[14] X. Zhang, R.M.Haralick and V. Ramesh, "Corner detection using MAP estimation," Manuscript submitted to CVPR'94.