Performance Characterization of Vision Algorithms

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

ABSTRACT

In order to design vision systems which work, a sound engineering methodology must be utilized. In the systems engineering approach, a complex system is divided into simple subsystems and from the input/output characteristics of each subsystem, the input/output characteristics of the total system can be determined. Machine vision systems are complex, and they are composed of different algorithms applied in sequence. Determination of the performance of a total machine vision system is possible if the performance of each of the subpieces, i.e. the algorithms, is given. The problem, however, is that for most algorithms, there is no performance characterization which has been established and published in the research literature.

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 imperfections of the input data. This paper illustrates how random perturbation models and propagation of random errors can be set up for a vision algorithm involving edge detection, edge linking, arc segmentation, and line fitting. The paper also discusses important dimensions that must be included in the performance characterization of any vision module performing a parametric estimation such as object pose, curve fit, or edge orientation estimation. Finally, we outline a general parametric model having three components: a relational model; a noise model; and a computational estimation model.

1. Introduction

Machine vision systems are complex and they are composed of different algorithms applied in sequence. Determination of the performance of a total machine vision system is possible if the performance of each of the subsystems, i.e. the algorithms, is given. The problem, however, is that for most algorithms, there is no performance characterization which has been established and published 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 imperfections of the input data.

Now we are thrown into an immediate problem. It is typically the case that an algorithm changes the data unit. For example, an edge-linking process changes the data from the unit of pixel to the unit of a group of pixels. An arc segmentation/extraction process applied to the groups of pixels produced by an edge linking process produces fitted curve segments. This unit change means that the representation used for the random variation of the output data set may have to be entirely different than the random variation used for the input data set. In our edge-linking/arc extraction example, the input data might be described by the false alarm/misdetection characteristics produced by the preceding edge operation, as well as the random variation in the position and rotation of the correctly detected edge pixels. The random variation in the output data from the extraction process, on the other hand, must be described in terms of fitting errors (random variation in the fitted coefficients) and segmentation errors.

The representation of the segmentation errors must be natural and suitable for the input of the next process in high-level vision which might be a model-matching process, for example. What should this representation be to make it possible to characterize the identification accuracy of the model matching as a function of the input segmentation errors and fitting errors? Questions like these, have typically not been addressed in the machine vision literature. Until they are, analyzing the performance of a vision algorithm will be in the dark ages of an expensive experimental trial-and-error process. This paper illustrates how random perturbation models can be set up in a simple setting for a vision algorithm involving edge detection, edge linking, arc segmentation, and line fitting. The paper also discusses important dimensions that must be included in the performance characterization of any vision module performing a parametric estimation such as object pose, curve fit, or edge orientation estimation. Finally, we outline a general parametric model having three components: a relational model; a noise model; and a computational estimation model.

2. Background Discussion

Our position is that a proposed vision methodology or architecture should be logically or intellectually and experimentally compelling. That is, the reasons for doing things one way rather than another should not be chosen because of heuristics, but because there is some prior information which indicates that the choice is among the best choices that could be made in the given circumstances and best choices are those based on a methodology whose assumptions have been experimentally verified.

We must therefore ask: what is it that compels one to adopt a particular vision methodology or architecture? The answer to this question involves answering questions about representations, algorithms, and accuracy. An algorithm employed at any stage in the image analysis sequence employs a representation for the data with which it works. Then we must ask: what kinds of conditions exceed the limits of the representation? When is reality not covered by the representation? These will be the cases of sure failure. The algorithm itself is a numerical calculation. So we must ask under what conditions does it break down because there are singular computational instances in which the numerical calculation is not stable? Finally, the algorithm works with noisy data, data which has been perturbed from its ideal form. Therefore, the results of the algorithm will be perturbed from their ideal form too. To what degree will a perturbation of the input data affect the accuracy of the output, in a qualitative sense and in a quantitative sense?

The methodology should employ both black-box and white-box perspectives. In the black-box perspective, we care about the requirements relating to the computer vision task the black-box is supposed to solve, and we determine whether these requirements are met by an empirical performance analysis. In the white-box perspective, the box is examined from the inside out. We ask whether it provides guaranteed answers under a given set of well-defined assumptions. We establish the guarantee by theory. Then we ask whether the assumptions are consistent with the reality the box is supposed to handle. We establish this consistency by an experimental reality-assumption validation test.

Everything that is said here about the box — white or black — meaning an individual algorithm that is a part of a total vision system can likewise be said for the total vision algorithm.

Section 3 of the paper gives a discussion of an example mid-level vision module for which the random perturbation process of the previous low-level vision stage involves units that are different from the units produced by the output of the mid-level vision process, so that different random perturbation processes must be employed for characterizing the input and output randomness. The example illustrates how, with some thought, it is possible to handle the situation, how the parameters of the random perturbation process can be estimated, and how the analysis that propagates the characterization of the random perturbation of the input to the random perturbation of the output can be done.

Section 4 gives a discussion of the kinds of performance characterization issues that must be handled for any vision module performing a parametric estimation. Vision modules doing 2D or 3D pose estimation, curve fitting, and edge orientation are just a few examples of such kinds of vision modules. Important issues of robustness, reliability, and error propagation are discussed in Section 4.

3. Example Discussion

In sections 1 and 2 of the paper, we outlined some of the fundamental questions of performance characterization. In this section, we change our perspective from the abstract to the concrete to illustrate how one of the random perturbation models can be set up in the simple setting described here for a vision algorithm involving edge detection, edge linking, arc segmentation, and line fitting. The example focuses on edge-linking, but there is no implied limitation that the proposed methodology cannot be employed to handle more complex vision algorithms involving perspective projection, matching, high-level control, and complex objects or scenes. We only employ this example for its ease and convenience. Note that the input random perturbation model differs from the output random perturbation model. This happens whenever the data structure of the output differs from the data structure of the input.

3.1 Random Perturbations at the Edge-Linking/Line-Fitting Stage

In this section we discuss the nature of the input data and the output data seen by an edge-linking mid-level-vision module and then proceed to model the random perturbation of the input and give details of how to estimate the model parameters from first principles. This discussion is illustrative of what can be done at any step of the vision process.

Let us assume that there is a step edge which is corrupted with noise. Edge detection may mislabel a pixel by assigning the non-edge label to edge pixels and assigning the edge label to non-edge pixels. So, for edge detection, the output random perturbation model can be specified by the misdetection and false alarm rates that are specified for the pixel as the data structure. A more involved edge detection model could take into account spatial correlations or dependencies in pixel misdetection and false alarm rates.

The edge-linking stage makes use of some information like spatial proximity and gradient direction to link the edge-labeled pixels. The output data structure produced by the edge-linking process is set of line or arc segments. However, if the noise level is quite high, the linking process will leave gaps between successive arcs, and whatever gap-filling process it employs will often be unable to close all the gaps between the edge segments. Therefore edge-linking leaves a collection of line segments with gaps between them. From this discussion one can view the input to the line-fitting stage as a collection of pixels belonging to short line segments. The entity that is the ideal input is the entire collection of pixels which belong to the line. It is reasonable for the random perturbation process that breaks these ideal long line segments into a collection of short line segments to possess the following characteristics:

- Long model line segments get broken up more than short model line segments. Th number of breaks is proportional to the length of the line segment, contrast across th edge being assumed constant.
- Short isolated line segments are more likely caused by correlated noise in the image.
- The probability of missing a line segment is higher for smaller length lines.
- Orientation estimates for smaller line segments may not be close to the true value. Thi is mainly because of the number of points involved in the estimation and the length o the line.
- The gap lengths between collinear image line segments arising from the same model line segment are small.
- Two model line segments intersecting at a very obtuse angle may give rise to a single linked line segment in the image.

All the above statements applies to curve segments as well. In the next section, we proceed to the mathematical characterization of the random perturbation process which breaks lines or curves.

The tasks that have to be accomplished are:

1) to state a random perturbation model suitable for characterizing random line lengths;
2) to use this random perturbation model to determine the probability density function for the interval length between two breaks;
3) to show how the free parameters of this random perturbation model relate to the statistical parameters of the previous process, which in this case is edge detection;
4) to show how the free parameters of this random perturbation model can be estimated from suitably labeled sample imagery; and
5) to show how the random perturbation of the input to a gap-filling process that is part of an edge-linking procedure can be propagated to the random perturbation process that characterizes the output of the gap-filling process.

3.2 Perturbation Model for Lines/Curves

Suppose that we have a curve C which is defined parametrically as:

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

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

where s is the arc length, $\phi_1, \ldots \phi_M$ are the given basis functions, $\alpha_1, \ldots, \alpha_M$ and $\beta_1, \ldots, \beta_M$ are the unknown coefficients. The parametrization is assumed to be defined in the interval $[0, L]$, where L is the length of the curve. Given such a curve, we describe a process which generates broken curve segments. A random variable X is exponentially distributed with parameter λ if:

$$\text{Prob}(X = x) = \lambda \exp^{-\lambda x}. \tag{1}$$

It should be noted that the mean value of X is $\frac{1}{\lambda}$.

We can model the breaks by assuming that the curve 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.

Assuming that we are given the interval $[0, L]$ and rate parameters λ_1 and λ_2 for the segment lengths and the gap lengths, the random perturbation process generates the collection of line/curve segments as follows:

- Start from the left end point of the interval (this corresponds to $s = 0$) and choose a value s_1 from an exponential distribution with rate parameter λ_1. The sequence of (r,c)'s in the interval $[0, s_1]$ belong to one curve / line segment. If L is such that the chosen value s_1 is greater than L then there are no breaks in the entire interval and we terminate our generation process.
- Given s_1, we choose a value s_2 from an exponential distribution with rate parameter λ_2. The sequences of (r,c)'s in the interval $[s_1, s_1 + s_2]$ are pixels that do not belong to the output segment. That is, these pixels form the gap or the break between curve segments. If $(s_1 + s_2)$ is greater than L, we terminate and we assume that there are no breaks.
- We continue in this fashion until the sum $\sum_i s_i$ is greater than L which is the condition for termination.

3.3 Probability Density Function of Interval between Two Breaks

Let X denote the random variable giving the distance between two successive starting points of line segments generated by the perturbation process described in the previous section. In this section we derive the expression for the probability density function for the random interval X. We then derive the expression for the mean number of breaks in a line/curve of length L in the subsequent section. Let X_1 be an exponential random variable, $E(\lambda_1)$, with rate parameter λ_1. Let X_2 be an exponential random variable with rate parameter λ_2, $E(\lambda_2)$. Since $\frac{1}{\lambda_2}$ corresponds to the mean gap length, $\lambda_2 >> \lambda_1$. We know that: $X = X_1 + X_2$. The probability density function of X is therefore the convolution of the individual probability densities of X_1 and X_2. Therefore:

$$\text{Prob}(X = x) = \int_0^x \lambda_1 e^{-\lambda_1 y} \lambda_2 e^{-\lambda_2 (x-y)} dy.$$

Simplifying, we can show that:

$$\text{Prob}(X = x) = \frac{\lambda_1 \lambda_2}{\lambda_2 - \lambda_1} \left(e^{-\lambda_1 x} - e^{-\lambda_2 x}\right). \tag{2}$$

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

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

From this, we can determine that the mean number of breaks in a line is equal to the ratio of the length of the entire line to the mean value of line segment length.

3.4 Edge Segment / Gap Distributions / Arc Lengths

In this section, we derive expressions for the probability distributions for the gap lengths and edge segment lengths given sample image data. We assume that an edge operator is applied to an input image and pixels in the output image are labelled edge pixels according to some criterion. Often pixels are labelled edge pixels if the gradient magnitude at a pixel is greater than a specified threshold or if the contrast at a pixel is greater than a specified threshold. In general we assume that a pixel is labelled an edge pixel if $g(r,c) > T$ where $g(r,c)$ is the result of applying some filter on the input image values $f(r,c)$ and T is some threshold. That is:

$$O(r,c) = \begin{cases} 1 & \text{if } g(r,c) > T \\ 0 & \text{otherwise} \end{cases}.$$

Since $G = g(r,c)$ at a particular pixel (r,c) is a random variable dependent on the input image values, if we assume a noise model for the input then the functional form for the distribution of G can be determined from first principles. Given that $p(g)$ is the probability density function for G, its distribution function $P(g)$ is useful in describing our line breaking process. The probability that a particular pixel (r,c) is labelled an edge pixel is given by:

$$\text{Prob}(G > T) = 1 - P(G \leq T).$$

The probability that i adjacent pixels will be labelled edge pixels is given by:

$$\prod_i (1 - P(G_i \leq T))$$

where $G_1, \ldots, G_i$ are i.i.d random variables.

Assume that we have a model line of N pixel length. Let $p_i = 1 - P(G_i \leq T)$, denote the probability that the ith pixel along the line gets labelled as an edge pixel. Imagine that one starts from the left end of the line and walks until he/she encounters a break. Let q_{j+1} denote the probability that the $j+1$th pixel gets labelled as an non-edge pixel. The probability that the break will be at location $j, j+1$ is given by:

$$\prod_{i=1}^{j} p_i q_{j+1} \quad .$$

We can therefore write the expression for the density function for the length of the segment, $x_1(j)$, as:

$$Prob(X_1 = j) = S_1 \prod_{i=1}^{j} p_i q_{j+1} \quad .$$

Here S_1 is a scale factor which is equal to:

$$S_1 = \frac{1}{\sum_{j=1}^{\infty} \prod_{i=1}^{j} p_i q_{j+1}}$$

Note that from first principles we can derive the analytical expression for p_i and q_i. As shown in the next section G_i is related to the non-central chi-square distribution. The density function of p_i is given by $1 - cdf(G_i)$. Here cdf stands for the cumulative distribution function. Since we can derive the functional form for the density function of p_i we can find the Laplace transform of the density function and derive the expression for the mean number of breaks.

The mean length of edge segments, μ_l, would be given by:

$$\mu_l = \sum_{j=1}^{j=\infty} j x_1(j)$$

and the variance is given by:

$$\sum_{j=1}^{j=\infty} ((j - \mu_l)^2) x_1(j))$$

The analysis to obtain the gap distribution proceeds in a similar fashion. A particular pixel will be labelled a non-edge when $G \leq T$. This probability is given by: $P(G \leq T)$, and the probability that i adjacent pixels will be labelled non-edge pixels is given by:

$$\prod_i (P(G_i \leq T)).$$

Let $q_i = P(G_i \leq T)$. Then the gap length density function is given by $x_2(j)$:

$$Prob(X_2 = j) = S_2 \prod_{i=1}^{j} q_i p_{j+1}$$

where S_2 is equal to:

$$S_2 = \frac{1}{\sum_{j=1}^{\infty} \prod_{i=1}^{j} q_i p_{j+1}}$$

The mean gap length $\frac{1}{\lambda_2}$ will then be given by:

$$\frac{1}{\lambda_2} = \sum_{j=1}^{j=\infty} j x_2(j)$$

3.5 Maximum-Likelihood-Estimates for λ_1 and λ_2

In this section we derive the expression for the maximum likelihood estimate of the parameters λ_1 and λ_2 from empirical observation of a sample of broken lines or curves. For the purpose of this derivation, it is assumed that we are trying to estimate the parameters given the sequence of edge segment lengths and gap lengths. Let a line be broken into n segments and let x_i, x'_i give the ith segment width and the ith gap length respectively. Then the likelihood function is given by:

$$\prod_{i=1}^{n} \left(\lambda_1 \lambda_2 e^{-\lambda_1 x_i} e^{-\lambda_2 x'_i} \right).$$

The log-likelihood function is therefore:

$$nln(\lambda_1) + nln(\lambda_2) - \lambda_1 \sum_{i=1}^{n} x_i - \lambda_2 \sum_{i=1}^{n} x'_i.$$

Taking partial derivatives with respect to λ_1 and λ_2 and setting to zero gives:

$$\frac{1}{\lambda_1} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

and

$$\frac{1}{\lambda_2} = \frac{1}{n} \sum_{i=1}^{n} x'_i.$$

Thus the maximum likelihood estimate of $\frac{1}{\lambda_1}$ is the mean of the observed edge segment lengths. Similarly the maximum likelihood estimate of $\frac{1}{\lambda_2}$ is the mean of the observed gap lengths. Thus:

$$\hat{\lambda_1} = \frac{1}{\frac{1}{n} \sum_{i=1}^{n} x_i}$$

$$\hat{\lambda_2} = \frac{1}{\frac{1}{n} \sum_{i=1}^{n} x'_i}$$

3.6 Analysis of a Gap Filling Technique

Edge or gap filling can be thought of as a process that fills in gaps of lengths less than some threshold, say L. For this analysis we assume that we have a broken line segment with the gap length and line length parameters of λ_2 and λ_1, respectively. We assume that the line breaking process is a renewal process with event interval length density function as given in equation(2). The problem we solve in this section is to characterize the random perturbation of the output of a gap-filling process that simply fills gaps in the input of length less than L. Let q denote the probability that the gap length in the input is less than L. Then:

$$\begin{aligned} q &= \int_0^L \lambda_2 e^{-\lambda_2 y} dy \\ &= 1 - e^{-\lambda_2 L}. \end{aligned}$$

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

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

and the mean number of gaps filled will be:

$$\begin{aligned} \mu &= \sum_{i=0}^{n} i \binom{n}{i} q^i (1-q)^{n-i} \\ &= nq \\ &= n(1 - e^{-\lambda_2 L}). \end{aligned}$$

The mean number of gaps that are not filled is then given by:

$$ne^{-\lambda_2 L}.$$

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

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

$$p(1-p)^{i-1}.$$

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

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

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

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

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

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

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

$$\begin{aligned} M_f(t) &= \left(\frac{\lambda_1 p}{\lambda_2}\right) [\lambda_2 t - (1 - e^{-\lambda_2 t})] \\ &= \lambda_1 p t - \left(\frac{\lambda_1 p}{\lambda_2}\right) (1 - e^{-\lambda_2 t}). \end{aligned}$$

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

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

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

4. Errors, Error Propagation, Robustness, and Reliability

This section describes important dimensions that must be included in the performance characterization of any vision module performing a parametric estimation such as object pose, curve fit, or edge orientation estimation. There are seven basic issues:

1) If the input data/imagery has no random perturbations from the idealized model, how far from the true or correct result is the answer provided by the vision module?
2) If the input data/imagery has small random perturbations affecting most of the data, how well can these perturbations be reasonably characterized by a variance statistic or a covariance matrix?
3) How do the small random perturbations which affect most of the data propagate through the vision module and affect the output answer?
4) How can those very large random perturbations, which might affect a small portion of the data, be characterized?
5) How do the very large random perturbations, which affect a small portion of the input data, propagate through the vision module and produce a random perturbation in the output result, i.e., how robust is the vision module?

6) How statistically efficient is the computation of the vision module? What is the ratio of a measure of the error of the output result using input data having both large and small random perturbations to the measure of the error of the output result using input data having only small random perturbations?

7) How reliable is the answer provided? Does the vision module expect that the input data provided to it meet the model assumptions it used in processing the data and therefore believes the output it produced to be reliable? Or does it have a reason for believing that the input data did not meet the assumptions required by the processing algorithm and the answer it produced cannot, therefore, be believed?

4.1 A General Parametric Model

To answer the questions raised in the beginning of this section, we must pose them in the setting of a general model for parameter estimation which is what a substantial fraction of vision modules do. The general model we explore here has three components: a relational model; a noise model; and a computational estimation model. The relational model specifies the relationships between the unobserved and unknown parameters θ and the unobserved and unknown quantities x_i whose noisy realizations will be observed. We can compactly write this as

$$f_i(x_i, \theta) = 0, \quad i = 1, \ldots, I.$$

The noise model specifies the relationship between the observed noisy realization $\widehat{x_i}$ and the unobserved and unknown x_i. We can compactly write this as

$$\widehat{x_i} = x_i + \eta_i, \quad i = 1, \ldots, I$$

where η_i is the additive random noise. The computational estimation model specifies how the noisy realization $\widehat{x_i}$ is used to calculate an estimate $\hat{\theta}$ of θ. We can compactly write this as $\hat{\theta} = g(\widehat{x_i}, \ldots, \widehat{x_I})$.

4.2 Exterior Orientation

θ represents the unknown translation and rotation parameters that transform the object coordinate system to the camera coordinate system. Each x_i is the ideal noiseless 2D perspective projection of the i^{th} 3D object vertex. The function f_i gives the constraint between the rotation and translation parameters, the 3D vertex position, and the ideal 2D perspective projection of the i^{th} vertex:

$$f_i(x_i, \theta) = 0, \quad i = 1, \ldots, I.$$

The 3D vertex positions are built implicitly into the relation by having the function f be indexed by i. Each $\widehat{x_i}$ represents the noisy observation of the 2D perspective projection of the i^{th} vertex. The noisy observation $\widehat{x_i}$ of the 2D perspective projection comes from an additive perturbation η_i of the idealized 2D perspective projection x_i, $\widehat{x_i} = x_i + \eta_i, i = 1, \ldots, I$. Then given the noisy observed 2D perspective projections $\widehat{x_i}, \ldots, \widehat{x_I}$ and the corresponding known 3D positions of the object vertices in the object coordinate system, an estimate $\hat{\theta} = g(\widehat{x_1}, \ldots, \widehat{x_I})$ of the unknown rotation and translational parameters is determined. Here the known 3D position of the object vertices in the object coordinate system are built into the function g and are not expressed explicitly in the argument of g.

4.3 Curve Fitting

θ represents the unknown parameters of the curve whose form is known. Each x_i is the ideal noiseless position of a point on the curve and, therefore, satisfies $f(x_i, \theta) = 0, i = 1, \ldots, I$. Each $\widehat{x_i}$ is an observed noisy realization of the point x_i which lies on the curve; $\widehat{x_i} = x_i + \eta_i, i = 1, \ldots, I$, where η_i is the additive random perturbation affecting the i^{th} point. The unknown curve parameters θ are then estimated by $\hat{\theta}$ on the basis of the noisy observations; $\hat{\theta} = g(\widehat{x_i}, \ldots, \widehat{x_I})$.

4.4 Edge Detection

θ represents the unknown parameters (position, orientation, contrast) of an edge pixel. x represents the n-tuple of all the ideal brightnesses of pixels in the neighborhood about the given edge pixel. The edge parameters and the ideal neighborhood brightnesses are related by $f(x, \theta) = 0$. The observed noisy neighborhood brightnesses $\hat{x}$ are related to the ideal brightnesses x by $\hat{x} = x + \eta$, where η is the random noise. On the basis of the observed $\hat{x}$, the edge detection operator calculates $\hat{\theta}$ which is an estimate of the edge parameters θ; $\hat{\theta} = g(\hat{x})$.

4.5 The Questions

Now we ask in a more precise way the questions we asked in the beginning of this section relative to the general model

$$f_i(x_i, \theta) = 0, \quad i = 1, \ldots, I;$$

$$\widehat{x_i} = x_i + \eta_i, \quad i = 1, \ldots, I;$$

$$\hat{\theta} = g(\widehat{x_1}, \ldots, \widehat{x_I}).$$

Question 1: If the input data is noiseless, how far from the true or correct result is the estimate $\hat{\theta}$? That is, if x_i satisfies $f_i(x_i, \theta), i = 1, \ldots, I$ and $\hat{\theta} = g(x_1, \ldots, x_i)$, then how far is $||\theta - \hat{\theta}||$ from zero?

Question 2: If η_i is a small random perturbation and not a contaminating outlier perturbation, then is it reasonable to model the statistics of η_i by its covariance matrix Σ_i?

Question 3: What is the covariance matrix of $\hat{\theta}$?

Assuming $\theta = g(x_1, \ldots, x_I)$, and that $\eta_1, \ldots, \eta_I$ are independent and small enough that a first order multi-dimensional Taylor series expansion of g around $(x_1, \ldots, x_I)$ is accurate,

$$\hat{\theta} = g(\widehat{x_1}, \ldots, \widehat{x_I}) = g(x_1 + \eta_1, \ldots, x_I + \eta_I)$$
$$= g(x_1, \ldots, x_I) + \sum_{i=1}^{I} \left(\frac{\partial g}{\partial x_i} \right)' \eta_i$$
$$= \theta + \sum_{i=1}^{I} \left(\frac{\partial g}{\partial x_i} \right)' \eta_i.$$

Hence, we can compute the covariance matrix of $\hat{\theta}$ by

$$\Sigma = E[(\hat{\theta} - \theta)(\hat{\theta} - \theta)']$$
$$= E\left[\sum_{i=1}^{I} \left(\frac{\partial g}{\partial x_i} \right)' \eta_i \left(\sum_{j}^{I} = 1 \left(\frac{\partial g}{\partial x_j} \right)' \eta_j \right)' \right]$$
$$= E\left[\sum_{i=1}^{I} \left(\frac{\partial g}{\partial x_i} \right)' \sum_{j=1}^{I} \eta_i \eta_j' \left(\frac{\partial g}{\partial x_j} \right) \right]$$
$$= \sum_{i=1}^{I} \left(\frac{\partial g}{\partial x_i} \right)' \Sigma_i \left(\frac{\partial g}{\partial x_i} \right).$$

Question 4: How can those η_i which are outliers be characterized? Those η_i can be modeled as coming from a distribution having large or thick tails.

Question 5: What is the influence of one or many outlier η_i's on the estimate $\hat{\theta}$? Since an outlier η_i can be a large value, we can determine the influence by examining

$$\epsilon(\lambda) = \max_{\substack{\eta_i \\ \|\eta_i\| = \lambda}} \|\theta - g(x_1, \ldots, x_{i-1}, x_i + \eta_i, x_{i+1}, \ldots, x_I)\|.$$

If $\epsilon(\lambda)$ grows unbounded as λ grows unbounded, a single outlier η_i can influence the resulting estimate in an arbitrarily large way.

To determine the influence of multiple outlier η_i's, we can determine the breakdown point, which is the smallest fraction of the I noise perturbations that have to grow arbitrarily large in order to cause the estimate $\hat{\theta}$ to be arbitrarily different from θ.

Question 6: What is the statistical efficiency of the estimate? Suppose that $\eta_1, \ldots, \eta_J$ are small perturbations and $\eta_J + 1, \ldots, \eta_I$ are outlier perturbations. The best we can hope for the estimate $\widehat{\theta_I}$ produced from the general model

$$f_i(x_i, \theta) = 0, \quad i = 1, \ldots, I$$

$$\widehat{x_i} = x_i + \eta_i, \quad i = 1, \ldots, I$$
$$\widehat{\theta_I} = g_I(\widehat{x_1}, \ldots, \widehat{x_I})$$

involving the good point and the outliers is for this estimate not to be too different from the estimate produced from the general model

$$f_i(x_i, \theta) = 0, \quad i = 1, \ldots, J$$

$$\widehat{x_i} = x_i + \eta_i, \quad i = 1, \ldots, J$$

$$\widehat{\theta_J} = g_J(\widehat{x_1}, \ldots, \widehat{x_J}).$$

which involves only the good points. So we can examine $E[||\widehat{\theta_I} - \widehat{\theta_J}||^2]$ and examine the ratio trace $\Sigma_{\widehat{\theta_J}}$/ trace $\Sigma_{\widehat{\theta_I}}$ as a measure of the efficiency.

Question 7: How reliable is the estimate? To answer this questions, we reason as follows:

Once an estimate $\hat{\theta}$ has been calculated, estimates $\widehat{\hat{x}_i}$ for the ideal quantization x_i can be determined by

$$\widehat{\hat{x}_i} = \underset{f(z,\hat{\theta})=0}{\operatorname{argmin}_z} ||z - \widehat{x_i}||.$$

The difference between the observed $\widehat{x_i}$ and the estimates $\widehat{\hat{x}_i}$ are the residuals, $r_i = \widehat{x_i} - \widehat{\hat{x}_i}$. It is important to determine the extent to which a noise perturbation η_i affects the residual r_i. Those points for which a significant fraction of the magnitude of the noise perturbation results in the magnitude of r_i are points for which errors in the observed $\widehat{x_i}$ caused by η_i are detectable. Points for which only an insignificant fraction of the magnitude of the noise perturbation results in the magnitude of r_i are points for which errors in the observed $\widehat{x_i}$ caused by η_i are not detectable. Situations having points associated with undetectable errors are precisely situations in which the reliability is suspect.

5. Conclusion

In this paper we illustrated how random perturbation models and propagation of random errors can be set up for a vision algorithm involving edge detection, edge linking, arc segmentation and line fitting. We also discussed important dimensions that must be included in the performance characterization of any vision model performing a parametric estimation. Almost all existing vision algorithms can be categorized into an algorithm class based on the type of input information they use and the output they produce. For example, there are many edge linkers and line finders that use directional information in conjunction with the positional information of line segments. These algorithms are very similar in principle and hence their performance could be analyzed by formulating a common algorithmic model. In our approach we are currently investigating the performance of classes of these algorithms by using analytical models. Since the outputs obtained from a particular algorithm for various inputs depend on the algorithm parameters chosen, there is a relationship between the algorithm parameters and the performance of the algorithm. We are interested in building a system that would use these relationships to select the optimal algorithm parameters for a vision sequence.