Error Propagation for Computer Vision Performance Characterization

Qiang Ji
Computer Science Department
University of Nevada at Reno

Robert M. Haralick
Department of Electrical Engineering
University of Washington

Abstract *This paper introduces error propagation techniques for analytically propagating positional image error to the estimated quantity produced by a vision algorithm. We focus on vision algorithms whose input and output are related either explicitly or implicitly. We then demonstrate the utilities of the error propagation techniques for a specific vision problem: curve-fitting.*
Keywords: Error propagation, performance characterization, computer vision

1 Introduction

Each computer vision problem begins with noisy images. The noises in the images may result from digitization, projection, sensor noises, etc. The accumulated effect of these noises induces locational error to each pixel in the image. In the subsequent vision algorithm, the locational error will be carried over through each vision step up to the final result. Compounded by additional error introduced by each intermediate vision step, the output of a vision algorithm is often uncertain. The uncertainties with the output of a vision algorithm determine the performance of the algorithm. Take the industrial inspection for example, the precision of a vision system largely depends on the uncertainties associated with the final measurements. Development of the best inspection algorithm requires understanding how the uncertainty due to perturbation affecting the input images propagates through different algorithmic steps and results in a perturbation on the output measurements. This means that we must propagate image error through each intermediate vision step up to the final output to characterize the performance of the vision algorithm.

The problem of error propagation is a fundamental issue in computer vision. We encounter this problem in different disguises. This problem occurs often in many applications of computer vision and can be considered of great practical importance [3]. In this paper, we discuss the problem of analytically computing the uncertainty of output parameters that are functions, either explicit or implicit, of some input measurements. We focus on error propagation for vision algorithms that obtain its output by minimizing a scalar criterion function, which is employed by many vision algorithms.

2 Image Error and Estimation

Understanding relationship between input perturbation and output perturbation requires to understand the nature of input perturbation and its distribution. Input to a vision algorithm consists of an image or image points. Image is noisy and is subject to various image errors.

Errors are introduced to an image from a variety of sources. Image errors may be classified into systematic errors and random errors. Systematic errors may include errors caused by lens distortion, perspective projection, etc. For error propagation purpose, we assume systematic errors are small or have been accounted for. Sources of random image errors may include illumination, spatial quantization, sensor position, and feature extraction techniques. These errors collectively cause intensity and positional inaccuracies to image points. Because of random uncertainties, as evidenced by the small differences between observations of the same image point, it is customary to model the random positional inaccuracies by means of a stochastic model.

Let $\hat{X}_n = (\hat{x}_n, \hat{y}_n)$ be the perturbed but observed pixel coordinates of an image point and $X_n = (x_n, y_n)$ be the ideal (without perturbation) but unknown pixel coordinates of the same image point. It is assumed that a small additive perturbation ΔX_n added to X_n leads to $\hat{X}_n$, i.e.,

$$\hat{X}_n = X_n + \Delta X_n \tag{1}$$

where ΔX_n represents the additive input perturbation. ΔX_n can be quantitatively characterized by its covariance matrix $\Sigma_{\Delta X_n}$. A simple but commonly used form for $\Sigma_{\Delta X_n}$ is

$$\Sigma_{\Delta X_n} = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix} \tag{2}$$

where σ_x^2 and σ_y^2 represent image perturbations in x and y directions. If we assume that the image perturbation in x and y directions are identical and independent, the noise model can be further simplified to

$$\Sigma_{\Delta X_n} = \sigma^2 I \tag{3}$$

where $\sigma_x^2 = \sigma_y^2 = \sigma^2$ and I is an identity matrix.

A more reasonable model is to assume that the uncertainty of an image point is along the direction of the image gradient at this point. This yields the following noise model

$$\begin{pmatrix} \hat{x}_n \\ \hat{y}_n \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \xi_n \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \tag{4}$$

where θ is image gradient direction at point (x_n, y_n) and ξ_n represents the random spatial perturbation with a variance of σ^2. This perturbation model leads to the following form for $\Sigma_{\Delta X_n}$

$$\Sigma_{\Delta X_n} = \sigma^2 \begin{pmatrix} \cos^2\theta & \sin\theta\cos\theta \\ \sin\theta\cos\theta & \sin^2\theta \end{pmatrix} \tag{5}$$

Graphically, $\Sigma_{\Delta X_n}$ can be represented by an ellipse with major and minor axes being the eigenvectors of $\Sigma_{\Delta X_n}$.

While the proposed noise models do not necessarily represent all perturbations, there are many visions algorithms where these models may hold or hold approximately.

For most vision algorithms, image pixels that participate in the computation are usually edge points or corner points. Most feature points are detected via a curve-fitting process (e.g., line fitting for corners and ellipse fitting for ellipse points). σ^2 can therefore be estimated from the residual fitting errors.

Specifically, let f be the curve function (either line or ellipse), Θ be the ideal curve parameters, and $\hat{X} = \{\hat{X}_n | n = 1, 2, \ldots N\}$ be the observed coordinates of N image points, we want to perform a least-squares curve fitting to fit $\hat{X}$ to f. Then $\hat{\Theta}$, the least-squares estimate of Θ, is found by minimizing the sum of geometric distances

$$\epsilon^2 = \sum_{n=1}^{N} \frac{f^2(\hat{X}_n, \hat{\Theta})}{(\frac{\partial f(\hat{X}_n, \hat{\Theta})}{\partial X_n})^t (\frac{\partial f(\hat{X}_n, \hat{\Theta})}{\partial X_n})} \tag{6}$$

Linearizing $f(\hat{X}_n, \Theta)$ with respect to the ideal value X_n, leading to

$$f(\hat{X}_n, \Theta) = f(X_n, \Theta) + (\frac{\partial f(X_n, \Theta)}{\partial X_n})^t \Delta X_n$$

since $f(X_n, \Theta) = 0$, then $f(\hat{X}_n, \Theta)$ is distributed as $N(0, (\frac{\partial f_n}{\partial X_n})^t \Sigma_{\hat{X}_n} (\frac{\partial f_n}{\partial X_n}))$, where $f_n = f(X_n, \Theta)$. The distribution of $f(\hat{X}_n, \Theta)$ can be further simplified using equation 3

$$f(\hat{X}_n, \Theta) \sim N(0, \sigma^2 (\frac{\partial f_n}{\partial X_n})^t (\frac{\partial f_n}{\partial X_n}))$$

Approximately, it follows that

$$\epsilon^2 \sim \sigma^2 \chi^2_{N-p} \tag{7}$$

where p is the dimension of Θ. Hence, we have $\hat{\sigma}^2$, an estimate of σ^2 is obtained via

$$\hat{\sigma}^2 = \frac{\epsilon^2}{N-p} \tag{8}$$

$\hat{\sigma}^2$ can be used as an unbiased estimate of σ^2. This estimator is available for each curve-fitting (line or ellipse). Because there may exist multiple curves in an image, the average estimator taken over all curve-fittings may be used as a good and stable estimate of σ^2.

3 Relationships between Input and Output

A computer vision algorithm works with three kinds of objects: an input vector, an output vector, and a function that maps input to output. Let X be the ideal but unobserved input vector with a dimension of $N \times 1$. Added to this unobserved ideal unperturbed vector is an random vector ΔX of the same dimension. This leads to $\hat{X} = X + \Delta X$, the observed input vector. Let Θ represents the ideal parameters vector of dimension $K \times 1$ generated by the vision algorithm if ideal input X is given. Given $\hat{X}$, the vision algorithm outputs $\hat{\Theta}$ instead, which relates to Θ via $\hat{\Theta} = \Theta + \Delta\Theta$.

For a computer vision algorithm, the relationships between its input (either X or $\hat{X}$) and the output quantity it computes (either Θ or $\hat{\Theta}$) can be grouped into three categories: explicit relationship, implicit relationship, and neither explicitly nor implicitly related through an analytic form. In the first case, input and output of a vision algorithm are explicitly related through an analytic function. Standard error propagation can be used to perform error propagation. The results are summarized in section 4.1 which gives the covariance matrix of the output parameters as a function of the Jacobi matrix and the covariance matrix of the measurements. Computer vision problems such as calculation of curvature, gradients, vanishing points, and feature points fall into this category.

In the frequent case where the parameters are obtained by minimizing some criterion function, the covariance matrix of the output parameters are expressed as the Hessian matrix and the covariance matrix of the measurements. This result is summarized in section 4.2. This is basically an optimization problem. This relationship embodies a wide variety of computer vision problems that can be analytically formulated as an optimization problem (either linear or non-linear). These problems may include curve-fitting, feature extraction, camera calibration, pose estimation, 3D reconstruction, and motion estimation as outlined by Haralick [4].

For cases where input and output can not be related either explicitly or implicitly through an analytic form, a statistical re-sampling technique described by Cho et al [1] may be employed for numerically estimating the covariance matrix of the output parameters.

4 Error Propagation Techniques

In this section, we describe the techniques for analytically propagating error from input to output, where input and output are related either explicitly or implicitly. The error estimation and propagation techniques studied here represents an extension of the standard error propagation technique [2].

4.1 Error propagation with explicit function

Following the same notation as introduced in section 2, here input vector $\hat{X}$ explicitly relates to the output vector $\hat{\Theta}$ by a non-linear function F, i.e.,

$$\hat{\Theta} = F(\hat{X}) \tag{9}$$

Linearizing both sides of the above equation with respect to the ideal input X and output Θ yields

$$\Theta + \Delta\Theta = F(X) + (\frac{\partial F(X)}{\partial X})^t \Delta X$$

where $\Delta\Theta$ is the perturbation added to Θ due to ΔX. As a result,

$$\Delta\Theta = (\frac{\partial F(X)}{\partial X})^t \Delta X$$

Hence the perturbation on $\hat{\Theta}$, characterized by its covariance matrix $\Sigma_{\Delta\Theta}$, is

$$\begin{aligned} \Sigma_{\Delta\Theta} &= E[(\Delta\Theta)(\Delta\Theta)^t] \\ &= (\frac{\partial F(X)}{\partial X})^t \Sigma_{\Delta X} (\frac{\partial F(X)}{\partial X}) \end{aligned} \tag{10}$$

4.2 Covariance Propagation for Implicit Functions

In many cases, $\hat{\Theta}$ and $\hat{X}$ are not related through an explicit function but through an non-linear minimization function F, i.e., $\hat{\Theta}$ is determined by minimizing $F(\hat{X}, \hat{\Theta})$. Then covariance propagation from $\hat{X}$ to $\hat{\Theta}$ can be performed using Haralick's covariance propagation theory[4]. Covariance propagation can be performed in unconstrained case and unconstrained case. We summarize the results in each case

4.2.1 Covariance propagation for unconstrained minimization

In this case, $\hat{\Theta}$ is determined solely by minimizing $F(\hat{X}, \hat{\Theta})$ without subject to any constraints. The technique assumes that 1) the criterion function F to be minimized has finite second partial derivatives; 2) $F(\Theta, X) = 0$ for the ideal input and output parameters X and Θ; 3) the input and out perturbations are small and additive.

Let $g(X, \Theta) = \frac{\partial F(X,\Theta)}{\partial \Theta}$, then the perturbation on the calculated parameters $\hat{\Theta}$, as represented by its covariance matrix Σ_Θ, relates to the input perturbation $\Sigma_{\Delta X}$ by

$$\Sigma_{\Delta\Theta} = [(\frac{\partial g}{\partial \Theta})^t]^{-1} (\frac{\partial g}{\partial X})^t \Sigma_{\Delta X} (\frac{\partial g}{\partial X}) (\frac{\partial g}{\partial \Theta})^{-1} \quad (11)$$

4.2.2 Covariance propagation for constrained minimization

Under this case, $\hat{\Theta}$ is determined by minimizing $F(\hat{X}, \hat{\Theta})$ subject to constraints $S(\hat{\Theta}) = 0$. Introducing the Lagrange multipliers, the function to be minimized is

$$F(\hat{X}, \hat{\Theta}) + S^t(\hat{\Theta})\Lambda$$

where Λ is a vector of Lagrange multipliers.

Define g to be $g(X, \Theta) = \frac{\partial F(X,\Theta)}{\partial \Theta}$, then making the same assumptions as in unconstrained case, we obtain

$$\Sigma_{\Delta\Theta,\Delta\Lambda} = A^{-1} B \Sigma_X B^t A \quad (12)$$

where

$$A = \begin{pmatrix} \frac{\partial g}{\partial \Theta} & \frac{\partial S}{\partial \Theta} \\ (\frac{\partial g}{\partial \Theta})^t & 0 \end{pmatrix} \quad B = -\begin{pmatrix} (\frac{\partial g}{\partial X})^t \\ 0 \end{pmatrix}$$

$\Sigma_{\Delta\Theta}$ can be obtained from the first $K \times K$ submatrix of $\Sigma_{\Delta,\Lambda}$.

For both the unconstrained and constrained cases, all functions are evaluated at ideal Θ and X. In practice, X and Θ are not available. We can obtain an estimate of the covariance matrix by replacing X with $\hat{X}$ and Θ with $\hat{\Theta}$, which is obtained from the minimization procedure.

5 Covariance Propagation for Curve-fitting

As part of the application of the error propagation techniques introduced in the previous section, this section demonstrates how to apply them to a very important computer vision problem: curve-fitting.

Least-squares curve fitting refers to determining the free parameters Θ of an analytical curve $F(x, y, \Theta) = 0$ such that the curve is the best fit to a set of points $(\hat{x}_n, \hat{y}_n)$ in the least-squares sense. A *best* fit is defined as a fit that minimizes the sum of squares of the geometric distances [5] as defined by

$$\epsilon^2 = \sum_{n=1}^{N} \frac{F^2(\hat{x}_n, \hat{y}_n, \Theta)}{(\frac{\partial F(\hat{x}_n,\hat{y}_n)}{\partial \hat{x}_n})^2 + (\frac{\partial F(\hat{x}_n,\hat{y}_n)}{\partial \hat{y}_n})^2} \quad (13)$$

Error propagation here relates the perturbations of points $(\hat{x}_n, \hat{y}_n)$ to the perturbation of $\hat{\Theta}$, the least-squares estimate of curve parameter Θ. Let $\Sigma_{\Delta X}$ and $\Sigma_{\Delta\Theta}$ be the covariance matrices of the observed points and estimated curve parameters. They are related via equation 11, where $g(X, \Theta)$ is defined as

$$g = \frac{\partial \varepsilon^2}{\partial \Theta} \quad (14)$$

From equation 14 and using $F(X_n, \Theta) = 0$, we obtain

$$\frac{\partial g}{\partial \Theta} = 2 \sum_{n=1}^{N} \frac{(\frac{\partial F_n}{\partial \Theta})(\frac{\partial F_n}{\partial \Theta})^t}{(\frac{\partial F_n}{\partial x_n})^2 + (\frac{\partial F_n}{\partial y_n})^2} \quad (15)$$

And let $X_n = (x_n\ y_n)^t$ and $X = (X_1 \ldots X_N)^t$, hence

$$\frac{\partial g}{\partial X_n} = 2\frac{(\frac{\partial F_n}{\partial X_n})(\frac{\partial F_n}{\partial \Theta})^t}{(\frac{\partial F_n}{\partial x_n})^2 + (\frac{\partial F_n}{\partial y_n})^2}$$

$$\frac{\partial g}{\partial X} = \left(\frac{\partial g}{\partial X_1} \quad \cdots \quad \frac{\partial g}{\partial X_n} \right)^t$$

It is clear from the above that

$$(\frac{\partial g}{\partial X})^t(\frac{\partial g}{\partial X}) = 2\frac{\partial g}{\partial \Theta} \tag{16}$$

Substituting the above relation and $\Sigma_{\Delta X} = \sigma^2 I$ into equation 11 yields

$$\Sigma_{\Delta\Theta} = 2\sigma^2[(\frac{\partial g}{\partial \Theta})^t]^{-1} \tag{17}$$

6 Covariance propagation for line, circle, and ellipses

6.1 Covariance Propagation for Line Fitting

Given a line expressed as

$$F(x, y, \Theta) = x\cos\theta + y\sin\theta + \rho \tag{18}$$

A least-squares line fitting amounts to finding the line parameter $\Theta = (\theta, \rho)$ that best fits a set of points $\hat{X} = (\hat{x}_1, \ldots, \hat{x}_n, \hat{y}_1, \ldots, y_N)$. Error propagation is concerned with estimating the perturbation of $\hat{\Theta}$, a least-squares estimate of Θ, given the perturbation with $\hat{X}$. $\hat{\Theta}$ is obtained by minimizing

$$\epsilon^2 = \sum_{i=1}^{N}(\hat{x}_i cos\hat{\theta} + \hat{y}_i \sin\hat{\theta} - \hat{\rho})^2 \tag{19}$$

Hence, $\frac{\partial g}{\partial \Theta}$ can be computed from equations 15 and 18 as follows

$$\frac{\partial g}{\partial \Theta} = 2\sum_{n=1}^{N}\begin{pmatrix} k_n^2 & -k_n \\ -k_n & 1 \end{pmatrix}$$

where $k_n = x_n \sin\theta - y_n \cos\theta$.

Let $u_k = \frac{\sum_{n=1}^{N} k_n}{N}$ and $S_k^2 = \sum_{n=1}^{N}(k_n - u_k)^2$, we have [1]

$$\Sigma_{\Delta\Theta} = \sigma^2 \frac{\begin{pmatrix} N & \sum k_n \\ \sum k_n & \sum k_n^2 \end{pmatrix}}{N\sum k_n^2 - \sum k_n \sum k_n}$$

[1] note $k_n^2 - k_n\mu_k = (k_n - \mu_k)^2$ and $\sum k_n^2 = S_k^2 + N\mu_k^2$

$$= \sigma^2 \begin{pmatrix} \frac{1}{S_k^2} & \frac{\mu_k}{S_k^2} \\ \frac{\mu_k}{S_k^2} & \frac{1}{N} + \frac{\mu_k^2}{S_k^2} \end{pmatrix} \tag{20}$$

6.2 Error Propagation for Circle Fitting

A circle can be represented by equation $F(x, y, \Theta) = (x - a)^2 + (y - b)^2 - R^2 = 0$, where (a, b) is the center of the circle and R is the radius of the circle. Given point scatter $\hat{X} = (\hat{x}_n, \hat{y}_n), n = 1, \ldots, N$, the least squares fitting amounts to estimating the parameter $\hat{\Theta} = (\hat{a}, \hat{b}, \hat{R})$ by minimizing the sum of squares of geometric distances as defined in equation 13.

Given $F(x, y, \Theta)$ as defined above, we have

$$\frac{\partial F_n}{\partial \Theta} = \begin{pmatrix} \frac{\partial F_n}{\partial a} \\ \frac{\partial F_n}{\partial b} \\ \frac{\partial F_n}{\partial R} \end{pmatrix} = -2\begin{pmatrix} x_n - a \\ y_n - b \\ R \end{pmatrix}$$

Hence using equation 15 leads to

$$\frac{\partial g}{\partial \Theta} = \frac{2}{R^2}\sum_{n=1}^{N}\begin{pmatrix} w_n^2 & w_n z_n & R w_n \\ w_n z_n & z_n^2 & R z_n \\ R w_n & R z_n & R^2 \end{pmatrix} \tag{21}$$

where $w_n = x_n - a$ and $z_n = y_n - b$. In the polar coordinate system, a circle is represented as

$$x_n = a + R\cos\alpha_n \quad y_n = b + R\sin\alpha_n$$

where α_n is the direction from circle center to point (x_n, y_n). $\frac{\partial g}{\partial \Theta}$ can therefore be reexpressed as

$$\frac{\partial g}{\partial \Theta} = 2\begin{pmatrix} \sum\cos^2\alpha_n & \sum\sin\alpha_n\cos\alpha_n & \sum\cos\alpha_n \\ \sum\sin\alpha_n\cos\alpha_n & \sum\sin^2\alpha_n & \sum\sin\alpha_n \\ \sum\cos\alpha_n & \sum\sin\alpha_n & N \end{pmatrix} \tag{22}$$

Substituting equation 22 to 17 yields the covariance matrix of the circle parameters.

$$\Sigma_{\Delta\Theta} = \sigma^2\begin{pmatrix} \sum\cos^2\alpha_n & \sum\sin\alpha_n\cos\alpha_n & \sum\cos\alpha_n \\ \sum\sin\alpha_n\cos\alpha_n & \sum\sin^2\alpha_n & \sum\sin\alpha_n \\ \sum\cos\alpha_n & \sum\sin\alpha_n & N \end{pmatrix}^{-1} \tag{23}$$

we can conclude from the above equation that 1) the variances of the estimated circle parameters do not depend on the circle radius; and 2) the variance of the estimated circle radius depends only on the number of points used.

6.3 Covariance Propagation for Ellipse Fitting

An ellipse may be expressed by the general conic equation

$$\begin{aligned} F(x, y, \Theta) &= A(x-a)^2 + 2B(x-a)(y-b) + \\ &\quad C(y-b)^2 - 1 = 0 \end{aligned} \tag{24}$$

To ensure the resulting fitting curve be an ellipse, parameters A, B, and C must be constrained such that $B^2 < 4AC$. Haralick [5] effectively proposed a way of implicitly incorporating this constraint in the fitting equation by working with a different set of parameters. According to this method, the parameters can be expressed as

$$\begin{pmatrix} A & B \\ B & C \end{pmatrix} = \begin{pmatrix} c & d \\ 0 & e \end{pmatrix}^t \begin{pmatrix} c & d \\ 0 & e \end{pmatrix} \tag{25}$$

It is clear from this relation that there exists c, d, and e, such that

$$\begin{pmatrix} A & B \\ B & C \end{pmatrix} = \begin{pmatrix} c^2 & cd \\ cd & d^2+e^2 \end{pmatrix} \tag{26}$$

In other words, there exist values of c, d, and e to make $B^2 - AC < 0$. One set of possible values for c, d, and e are:

$$A = c^2 \quad B = cd \quad C = e^2 + d^2$$

This means that we can make the fitting problem to be ellipse-specific by using the free parameters c, d, and e rather A, B, and C. With this perspective, the ellipse equation can be reexpressed as:

$$\begin{aligned} F(x, y, \Theta) &= c^2(x-a)^2 + 2cd(x-a)(y-b) + \\ &\quad (d^2+e^2)(y-b)^2 - 1 \end{aligned} \tag{27}$$

Ellipse fitting amounts to finding an ellipse $\hat{\Theta}$ that best fits a set of point $\hat{X}$.

Let $\Sigma_{\Delta X}$ and $\Sigma_{\Delta\Theta}$ be the covariance matrices of the observed points $\hat{X}$ and estimated ellipse parameter $\hat{\Theta} = (\hat{a}\ \hat{b}\ \hat{c}\ \hat{d}\ \hat{e})^t$, $\Sigma_{\Delta\Theta}$ may be computed from from equation 17, where $\frac{\partial g}{\partial \Theta}$ is computed from equation 15 as follows.

$$\frac{\partial g}{\partial \Theta} = 2\sum_{n=1}^{N} \frac{(\frac{\partial F_n}{\partial \Theta})(\frac{\partial F_n}{\partial \Theta})^t}{(\frac{\partial F_n}{\partial x_n})^2 + (\frac{\partial F_n}{\partial y_n})^2}$$

$$= 2\sum_{n=1}^{N} \begin{pmatrix} \frac{g_n^2}{g_n^2+h_n^2} & \frac{g_n h_n}{g_n^2+h_n^2} & \frac{-g_n l_n}{g_n^2+h_n^2} & \frac{-g_n k_n}{g_n^2+h_n^2} & \frac{-g_n w_n}{g_n^2+h_n^2} \\ \frac{g_n h_n}{g_n^2+h_n^2} & \frac{h_n^2}{g_n^2+h_n^2} & \frac{-h_n l_n}{g_n^2+h_n^2} & \frac{-h_n k_n}{g_n^2+h_n^2} & \frac{-h_n w_n}{g_n^2+h_n^2} \\ \frac{-g_n l_n}{g_n^2+h_n^2} & \frac{-l_n h_n}{g_n^2+h_n^2} & \frac{l_n^2}{g_n^2+h_n^2} & \frac{l_n k_n}{g_n^2+h_n^2} & \frac{l_n w_n}{g_n^2+h_n^2} \\ \frac{-g_n k_n}{g_n^2+h_n^2} & \frac{-k_n h_n}{g_n^2+h_n^2} & \frac{k_n l_n}{g_n^2+h_n^2} & \frac{k_n^2}{g_n^2+h_n^2} & \frac{k_n w_n}{g_n^2+h_n^2} \\ \frac{-g_n w_n}{g_n^2+h_n^2} & \frac{-h_n w_n}{g_n^2+h_n^2} & \frac{l_n w_n}{g_n^2+h_n^2} & \frac{k_n w_n}{g_n^2+h_n^2} & \frac{w_n^2}{g_n^2+h_n^2} \end{pmatrix}$$

where

$$\begin{aligned} g_n &= c^2(x_n-a) + cd(y_n-b) \\ h_n &= cd(x_n-a) + (d^2+e^2)(y_n-b) \\ l_n &= c(x_n-a)^2 + d(x_n-a)(y_n-b) \\ k_n &= c(x_n-a)(y_n-b) + d(y_n-b)^2 \\ s_n &= 2c(x_n-a) + d(y_n-b) \\ t_n &= c(x_n-a) + 2d(y_n-b) \\ d_n &= \frac{1}{g_n^2+h_n^2} \\ w_n &= e(y_n-b)^2 \\ F_n &= c^2(x_n-a)^2 + 2cd(x_n-a)(y_n-b) + \\ &\quad (d^2+e^2)(y_n-b)^2 - 1 \end{aligned}$$

Substituting the above equation to equation 17 yields the covariance matrix of the estimated ellipse parameters.

7 Covariance Propagation for Line fitting

In this section, we introduce a new technique [6] for error propagation with line fitting, where perturbation model for image point is not $\sigma^2 I$ as used in section 6.1. The input perturbation model follows equation 4. Following the same notations as in section 6.1, an estimate of Θ, is obtained by minimizing

$$\epsilon^2 = \sum_{i=1}^{N} (\hat{x}_i cos\hat{\theta} + \hat{y}_i \sin\hat{\theta} - \hat{\rho})^2 \tag{28}$$

$$g^{2\times 1} = \frac{\partial \epsilon^2}{\partial \Theta} = \begin{pmatrix} \frac{\partial \epsilon^2}{\partial \theta} \\ \\ \frac{\partial F}{\partial \rho} \end{pmatrix} \tag{29}$$

Then from equation 12, $\Sigma_{\Delta\Theta}$, the covariance matrix of line parameter $\hat{\Theta}$ can be computed from equation 11. From equation 29, we obtain

$$\frac{\partial g^{2\times 2}}{\partial \Theta} = \begin{pmatrix} \frac{\partial g}{\partial \theta} \\ \\ \frac{\partial g}{\partial \rho} \end{pmatrix} = \begin{pmatrix} \frac{\partial^2 F}{\partial \theta^2} & \frac{\partial^2 F}{\partial \theta \partial \rho} \\ \\ \frac{\partial^2 F}{\partial \rho \partial \theta} & \frac{\partial^2 F}{\partial \rho^2} \end{pmatrix}$$

and,

$$\frac{\partial g^{2N\times 2}}{\partial X} = \underbrace{\begin{pmatrix} \frac{\partial^2 F}{\partial \theta \partial x_1} & \frac{\partial^2 F}{\partial \rho \partial x_1} \\ \frac{\partial^2 F}{\partial \theta \partial y_1} & \frac{\partial^2 F}{\partial \rho \partial y_1} \\ \vdots & \vdots \\ \frac{\partial^2 F}{\partial \theta \partial x_N} & \frac{\partial^2 F}{\partial \rho \partial x_N} \\ \frac{\partial^2 F}{\partial \theta \partial y_N} & \frac{\partial^2 F}{\partial \rho \partial y_N} \end{pmatrix}}_{2N\times 2}$$

For the given perturbation model in equation 4, the input covariance matrix $\Sigma_{\Delta X}$ is given by

$$\Sigma_{\Delta X} = \sigma^2 \begin{pmatrix} d & \dots & 0 & 0 \\ 0 & d & \dots & 0 \\ \vdots & & & \\ 0 & 0 & \dots & d \end{pmatrix}_{N\times N}$$

where

$$d = \begin{pmatrix} \cos^2\theta & \sin\theta\cos\theta \\ \sin\theta\cos\theta & \sin^2\theta \end{pmatrix}$$

Define

$$k_n = x_n \sin\theta - y_n \cos\theta \qquad \mu_k = \frac{1}{N}\sum_{n=1}^{N} k_n$$

$$S_k^2 = \sum_{n=1}^{N} (k_n - \mu_k)^2$$

After algebraic operations and simplifications, we obtain

$$\sum\nolimits_{\Delta\Theta} = \begin{pmatrix} \sigma_\theta^2 & \sigma_{\theta\rho} \\ \sigma_{\theta\rho} & \sigma_\rho^2 \end{pmatrix} = \sigma^2 \begin{pmatrix} \frac{1}{S_k^2} & \frac{\mu_k}{S_k^2} \\ \frac{\mu_k}{S_k^2} & \frac{1}{N} + \frac{\mu_k^2}{S_k^2} \end{pmatrix} \quad (30)$$

Interestingly enough, this equation is the same as equation 20. This reveals that both noise models (equations 3 and 5) yield the same covariance matrix for the estimated line parameters.

Geometrically, k_n can be interpreted as the signed distance between a point (x_n, y_n) and the point on the line closest to the origin. As a result, S_k^2 represents the spread of points along the line. From equation 30, it is clear that with a larger S_k^2, i.e., points with larger spread along the line, we can obtain better fit as indicated with smaller covariance matrix. In addition, μ_k is the mean position of the points along the line. It acts like a moment arm. A larger μ_k, i.e. a longer moment arm, can induce more variance to the estimated $\hat{\rho}$. Further investigation of 30 reveals that σ_θ^2 is invariant to coordinate translation and rotation while σ_ρ^2 is only variant to coordinate rotation that changes μ_k.

8 Conclusions

In this paper, we introduce techniques for analytically propagating image error to characterize the performance of a vision algorithm by studying the uncertainty (reliability) of its output. Understanding error behavior of a vision algorithm often leads to finding techniques for improving accuracy and precision. Even if errors are inevitable, the knowledge of how reliable each vision algorithm is is indispensable in guaranteeing performance of the IU systems that use the algorithms. The proposed error propagation techniques may apply to a variety of vision algorithms. We are currently investigating the use of these techniques for important vision tasks like camera calibration, 3D reconstruction, and motion estimation.

References

[1] Kyujin Cho, Peter Meer, and Javier Cabrera. Performance assessment through bootstrap. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(11):1185–1198, 1997.

[2] H. F. Durrant-Whyte. Uncertain geometry in robotics. *IEEE Journal of Robotics and Automation*, 4(1):23–31, 1988.

[3] O. D. Faugeras. *Three Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.

[4] R. M. Haralick. Propagating covariance in computer vision. *Proc. of 12th IAPR*, pages 493–498, 1994.

[5] Robert M. Haralick and Linda G. Shapiro. *Computer and Robot Vision*, volume 2. Addison-Wesley Publishing Company, 1993.

[6] Qiang Ji and Robert M. Haralick. Corner detection with covariance propagation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(8), 1998.