Part 1: Statistics and Errors.

N. A. Thacker and P. Courtney.

- Methodology.
- Bayes Theorem.
- Maximum Likelihood.
- Common Probability Equations.
- Maximum Likelihood - Revisited.
- Non-Gaussian Errors.
- Dealing with Outliers.
- Non-Independent Measurements.
- Common Data Models.

Methodology.

- Vision algorithms must deliver information with which to make practical decisions regarding interpreting the data present in an image.

- All data will need an estimate of reliability.

- Probability is the only self-consistent computational framework for data analysis.

- The most direct form of information regarding an hypothesis is the posterior (often conditional) probability.

- There are several common models for statistical data analysis all of which can be related at some stage to the principle of maximum likelihood.

- Methods based on maximum-likelihood can provide the covariance (error) estimates we need for practical use of the data.

Bayes Theorem.

The basic foundation of probability theory follows from the following intuitive definition of conditional probability.

$$P(AB) = P(A|B)P(B)$$

In this definition events A and B are simultaneous and have no (explicit) temporal order we can write

$$P(AB) = P(BA) = P(B|A)P(A)$$

This leads us to a common form of Bayes Theory, the equation:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

which allows us to compute the probability of one event in terms of observations of another and knowledge of joint distributions.

Maximum Likelihood

Starting with Bayes theorem we can extend the joint probability equation to three and more events

$$P(ABC) = P(A|BC)P(BC)$$

$$P(ABC) = P(A|BC)P(B|C)P(C)$$

For n events with probabilities computed assuming a particular interpretation of the data (for example a model Y)

$$P(X_0 X_1 X_2 ... X_n|Y)P(Y) =$$

$$P(X_0|X_1 X_2 ... X_n Y)P(X_1|X_2 ... X_n Y) ...... P(X_n|Y)P(Y)$$

- Maximum Likelihood statistics involves the identification of the event Y which maximises such a probability. In the absence of any other information the prior probability $P(Y)$ is assumed to be constant for all Y.

- Even if the events were simple binary variables there are clearly an exponential number of possible values for even the first term in $P(XY)$ requiring a prohibitive amount of data storage.

- In the case where each observed event is independent of all others we can write.

  $$P(X_n|Y) = P(X_0|Y)P(X_1|Y)P(X_2|Y)...P(X_n|Y)$$

Dealing with Binary Evidence.

If we make the assumption that the event X_i is binary with probability $P(X_i)$ then we can construct the probability of observing a particular binary vector X as

$$P(X) = \Pi_i P(X_i)^{X_i} P(\tilde{X}_i)^{\tilde{X}_i}$$

or

$$P(X) = \Pi_i (P(X_i)^{X_i} (1 - P(X_i))^{(1-X_i)}$$

The log likelihood function is therefore

$$log(P) = \sum_i X_i log(P(X_i)) + (1 - X_i) log(1 - P(X_i))$$

This quantity can be or directly evaluated in order to form a statistical decision regarding the likely generator of X. This is therefore a useful equation for methods of statistical pattern recognition.

eg:

$$X = (0, 1, 0, ..., 1)$$

and

$$P(X) = (0.1, 0.2, 0.05, ..., 0.9)$$

Dealing with Data Distributions.

- The generation process for a histogram, making an entry at random according to a fixed probability, is described by the Poisson distribution.

  The probability of observing a particular number of entries h_i for an expected probability of p_i is given by

$$P(h_i) = exp(-p_i)\frac{p_i^{h_i}}{h_i!}$$

- For large expected numbers of entries this distribution approximates a Gaussian with

$$\sigma = \sqrt{p_i}$$

- The limit of a frequency distribution for an infinite number of samples and bins of infinitesimal width defines a probability density distribution.

These two facts allow us to see that the standard χ^2 statistic is appropriate for comparing two frequency distributions h_i and j_i for large measures.

$$-2\ log(P)\ =\ \chi^2 = \sum_i (h_i - j_i)^2/(h_i + j_i)$$

ie:

$$e^{-log(P)}\ =\ \Pi^i e^{-\chi_i^2/2}$$

Dealing with Functions.

If we now define the variation of the observed measurements X_i about the generating function Θ with some random error, the probability

$$P(X_0|X_1, X_2, ...X_N, \Theta, Y_0)$$

will be equivalent to $P(X_0|\Theta, Y_0)$.

Choosing Gaussian random errors with a standard deviation of σ_i gives

$$P(X_i) = A_i exp(\frac{-(X_i - f(\Theta, Y_i))^2}{2\sigma_i^2})$$

where A_i is a normalization constant. We can now construct the maximum likelihood function

$$P(\mathbf{X}|\Theta) = \Pi_i A_i exp(\frac{-(X_i - f(\Theta, Y_i))^2}{2\sigma_i^2})$$

which leads to the χ^2 definition of log likelihood

$$log(P) = \frac{-1}{2} \sum_i \frac{(X_i - f(y_i))^2}{\sigma_i^2} + const$$

- This expression can be maximized as a function of the parameters Θ and this process is generally called a least squares fit.

- Least squares fits are susceptible to fliers (outliers).

- The correct way to deal with these leads to the methods of robust statistics.

Maximum Likelihood - Revisited.

The most common approach for algorithm development is based on the idea of MAXIMUM LIKELIHOOD, which is derived from the joint probability:

$$P(Y\mathbf{X}) = (\Pi_i P(X_i|Y))P(Y)$$

Least squares (as we have seen) is derived from Probability theory on the assumption of indepedent Gaussian errors and that the prior probability of the model $P(Y)$ can be ignored.

such that:

$$\begin{aligned} log(P(\mathbf{X}|Y)) &= \sum_i log(P(X_i|Y)) \\ &= -\sum_i (X_i - f(i,Y))^2/\sigma_i^2 \end{aligned}$$

The best choice for Y is the one which maximises this likelihood. There are several key failings of

such an approach when used as the basis for machine vision algorithms.

Much research is thus directed (sometimes unknowingly) to overcoming these limitations.

Understanding what problems are being addressed and how is fundamental to making use of the results from other peoples research.

Non-Gaussian Errors.

Machine Vision is full of data that cannot be assumed to be from a Gaussian distribution.

There are two forms of problem:

- The error distribution may be relatively compact but badly skewed.

- There may be outliers caused by data "contamination".

The general technique for coping with the first problem is to transform the data to remove skewing.

eg:

$$D_x = f(x)$$

so we seek a function g which will give us

$$D_g(x) = const$$

using error propagation

$$D_g(x) \;=\; D_x \; dg/dx \;=\; f(f) \; dg/dx \;=\; const$$

ie: integrate the reciprocal of the error dependence:

$$g = \int \frac{const}{f(x)} dx$$

example. Stereo data.

$$z \;=\; fI/(x_l \;-\; x_r)$$

errors in $Pos(x, y, z)$ are badly skewed.

Attempting a LSF with these measures directly (eg for model location) is unstable due to large errors for large z.

However, errors on **disparity space**

$$Pos(x, y, fI/(\sqrt{2}z)$$

are uniform and can be used for fitting.

The technique can be considered as applying the inverse of error propagation (such as in image processing) in order to work back to a uniform distribution.

Dealing with Outliers.

This area of algorithm design is generally referred to as **Robust Statistics**. The simplest technique involves limiting the contribution of any data point to the total LSF ie:

$$-log(P) = \sum_i min((X_i - f(i, Y))^2/\sigma_i^2, 9.0)$$

The choice of 9.0 as the limit on the contribution is approximate and may depend on the problem.

This technique is not particularly good for methods which use derivatives during optimisation, as it introduces discontinuities which can introduce local minima.

Alternative involve replacing the Gaussian with a continuous distribution with long tails.

The most common of these is the double sided exponential.

$$-log(P) = \sum_i |(X_i - f(i, Y))/\sigma_i|$$

This is adequate for most applications.

More complex techniques which attempt to model slightly more realistic distributions can be found in the literature eg: Caucy distribution

$$P(X_i|Y) \;=\; \frac{1}{1 + (X_i \;-\; f(i,Y))/\sigma_i)^2/2}$$

so that our log probability is now

$$-log(P) \;=\; \sum_i log(1 \;+\; 1/2(X_i \;-\; f(i,Y))/\sigma_i)^2)$$

The price we pay for this is that, unlike standard least squares, such cost functions can rarely (never?) be optimised by direct solution so we have to use iterative techniques which take more time.

Non-Independent Measurements.

Under any practical circumstance the data delivered by a system may be correlated. It is then that we may need to preprocess the data to remove these correlations. This process is often called PRINCIPAL COMPONENT ANALYSIS.

We can define the correlation matrix

$$R = \sum_i (X_j - X_m) \otimes (X_j - X_m)$$

where X_j is an individual measurement vector from a data set and X_m is the mean vector for that set.

It can be shown that orthogonal (linearly independent) axes correspond to the eigenvectors V_k of the matrix R. Solution of the eigenvector equation

$$RV_k = \lambda_k V_k$$

The method known as Singular Value Decomposition (SVD) approximates a matrix by a set of orthogonal vectors W_l and singular values w_l.

$$R = \sum_l \frac{1}{w_l^2} W_l \otimes W_l$$

If we multiply both sides of the equation by one of these vectors W_k

$$RW_k = \sum_l \frac{1}{w_l^2} W_l \otimes W_l . W_k$$

we see that the singular vectors satisfy the eigen-vector equation with

$$\lambda_k = \frac{1}{w_k^2}$$

Identifying Correlations.

Correlation produces systematic changes in one parameter due to changes in another.

This can be visualised by producing a scatter-plot of the two variables $f(x, y)$.

In general for any two variables to be un-correlated knowledge of one must give no information regarding the other.

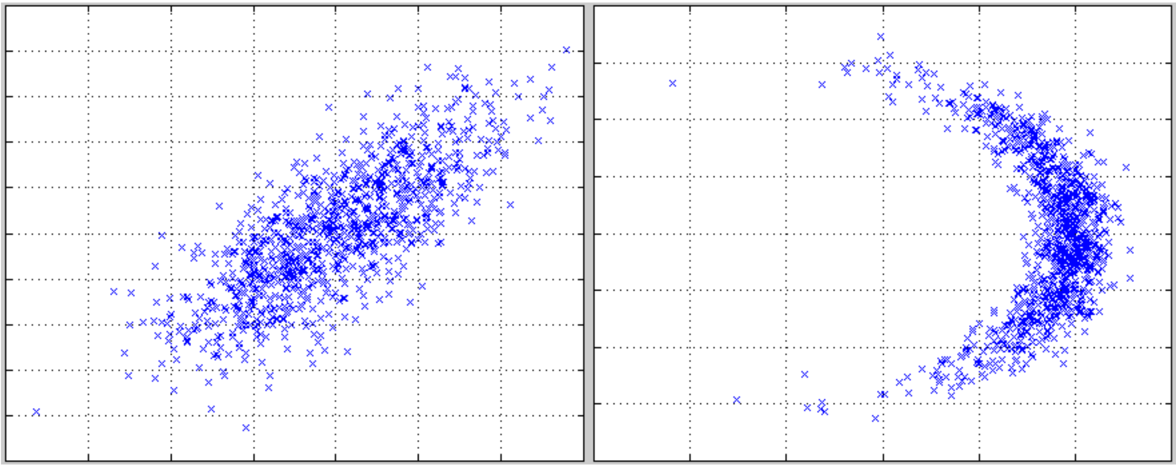
In terms of the scatter plot this means that the structure seen must be entirely modelable in terms of the outer-product of the two marginal distributions.

$$f(x, y) = f(x) \otimes f(y)$$

ie: decomposable.

Principle component analysis works by rotating the axes of the space to align along the axes of major variance of the data.

This may not necessarily de-correlate the data.



(Sozou et.al.1995)

Common Data Models.

Once good data models have been identified they can be used again in the design of new algorithms.

Data	Error Assumption
Images	Uniform random Gaussian + other
Histograms	Poisson sampling statistics
Edge features	Gaussian perpendicular to edge
Corner features	Circular (Eliptical) Gaussian
Line fits	Uniform Gaussian on end-points
3D Stereo data	Uniform in disparity space

Table 1 Standard error model assumptions.

Example; Object Location.

Demo: Pairwise geometric histogram model location (Ashbrook et. al. 1995)

Various transforms and algorithms can be used to achieve more uniform errors.

Some assumptions, such as uniform random errors on results from shape from optical flow methods, are not good models.

All models will probably need acceptance of outliers.

The old saying.... Junk in; junk out, could be better restated as:

Unknown statistical distributions in; Unknown statistical distributions out.

Part 2: Error Propagation and Monte-Carlo.

N. A. Thacker and P. Courtney.

- Covariance Estimation.
- Error Propagation.
- Image Processing Stability.
- Image Processing Errors.
- Image Arithmetic.
- Linear Filters.
- Histogram Equalisation.
- Monte-Carlo Techniques.

Covariance Estimation.

(Haralick 1996) For locally linear fit functions f we can approximate the variation in a χ^2 metric about the minimum value as a quadratic. We will examine the two dimensional case first, for example:

$$z = a + bx + cy + dxy + ex^2 + fy^2$$

This can be written as

$$\chi^2 = \chi_0^2 + \Delta X^T C_x^{-1} \Delta X \quad with \quad \Delta X = (x - x_0, y - y_0)$$

where C_x^{-1} is defined as the inverse covariance matrix

$$C_x^{-1} = \begin{vmatrix} u & v \\ w & s \end{vmatrix}$$

Comparing with the above quadratic equation we get

$$\chi^2 = \chi_0^2 + x^2 u + yxw + xyv + sy^2$$

where

$$a = \chi_0^2, b = 0, c = 0, d = w + v, e = u, f = s$$

Notice that the b and c coefficients are zero as required if the χ^2 is at the minimum.

Starting from the χ^2 definition using the same notation as previously.

$$\chi^2 = \frac{1}{2}\sum_i^N \frac{(X_i - f(y_i,a))^2}{\sigma_i^2}$$

We can compute the first and second order derivatives as follows:

$$\frac{\partial \chi^2}{\partial a_n} = \sum_i^N \frac{(X_i - f(y_i,a))}{\sigma_i^2}\frac{\partial f}{\partial a_n}$$

$$\frac{\partial^2 \chi^2}{\partial a_n \partial a_m} = \sum_i^N \frac{1}{\sigma_i^2}(\frac{\partial f}{\partial a_n}\frac{\partial f}{\partial a_m} - (X_i - f(y_i,a))\frac{\partial^2 f}{\partial a_n \partial a_m})$$

The second term in this equation is expected to be negligible giving

$$= \sum_i^N \frac{1}{\sigma_i^2}(\frac{\partial f}{\partial a_n}\frac{\partial f}{\partial a_m})$$

The following quantities are often defined.

$$\beta_n = \frac{1}{2}\frac{\partial \chi^2}{\partial a_n}$$

$$\alpha_{nm} = \frac{1}{2}\frac{\partial^2 \chi^2}{\partial a_n \partial a_m}$$

As these derivatives must correspond to the first coefficients in a polynomial (Taylor) expansion of the χ^2 function then,

$$C = \alpha^{-1}$$

And the expected change in χ^2 for a small change in model parameters can be written as

$$\Delta\chi^2 = \Delta a^T \alpha \Delta a$$

(Press et. al. 1988)

Error Propagation.

In order to use a piece of information $f(X)$ derived from a set of measures X we must have information regarding its likely variation.

If X has been obtained using a measurement system then we must be able to quantify measurement accuracy.

Then

$$\Delta f^2(X) = \nabla f^T C_X \nabla f$$

example 1: the Poisson distribution s

$$t = \sqrt{s}$$

then we can show, using a simplified form of error propagation for one parameter, that the expected variance on t is given by

$$\Delta t = \frac{\partial t}{\partial s} \Delta s$$

$$= \frac{-1}{2}$$

Thus the distribution of the square-root of a random variable drawn from a Poisson distribution with large mean will be constant.

example 2: Stereo Measurement

Using rectified images, the distance, Z between the feature and the camera plane can be found with the equation:

$$Z = \frac{fI}{x_1 - x_2}$$

where:
f is the focal length of the lenses
I is the inter-occular seperation
x_1 and x_2 are positions of the features on the epipolars
We can determine the sensitivity of Z with changes in x_1 and x_2 thus,

$$\Delta Z^2 = \left(\frac{\delta Z}{\delta x_1}\Delta x_1\right)^2 + \left(\frac{\delta Z}{\delta x_1}\Delta x_2\right)^2$$

where,

$$\frac{\delta Z}{\delta x_1} = -\frac{fI}{(x_1 - x_2)^2} \quad \text{and} \quad \frac{\delta Z}{\delta x_2} = \frac{fI}{(x_1 - x_2)^2}$$

Δx is the feature position error in the image and can be assumed to be equal in each image, so

$$\Delta x_1 = \Delta x_2 = \Delta x$$

Solving for ΔZ yields the result,

$$\boxed{\Delta Z = \frac{\sqrt{2} f I \Delta x}{(x_1 - x_2)^2} \quad \text{or w.r.t. } Z, \quad \Delta Z = \frac{\sqrt{2} Z^2 \Delta x}{f I}}$$

Image Processing Stability.

In simple image processing the requirements of an image processing algorithm may be purely to enhance the image for viewing.

But; the aim of advanced image processing to produce an image that makes certain information explicit in the resulting image values for automated data extraction.

eg: edge strength maps.

Generally, high values located over features of interest. The process which determines a good algorithm is its behaviour in the presence of noise, in particular does the resulting image give results which really can be interpreted purely on the basis of output value.

ie: is a high value genuine or just a product of the propagated noise.

In this lecture we will cover two ways of assessing algorithms: **Error Propagation** and **Monte-Carlo** techniques.

Image Processing Errors.

General Approach for Error Propagation (Recap).

$$\Delta f^2(X) = \nabla f^T C_X \nabla f$$

where ∇f is a vector of derivatives

$$\nabla f \; = \; (\frac{\partial f}{\partial X_1}, \frac{\partial f}{\partial X_2}, \frac{\partial f}{\partial X_3}, ...)$$

and $\Delta f(X)$ is the standard deviation on the computed measure

If we apply this to image processing assuming that images have uniform random noise then we can simplify this expression to

$$\Delta f_{xy}^2(I) = \sum_{nm} \sigma_{nm}^2 (\frac{\partial f_{xy}}{\partial I_{nm}})^2$$

ie: the contribution to the output from each independent variance involved in the calculation is added in quadrature.

Image Arithmetic.

We can drop the xy subscript as it is not needed.

Addition:

$$O = I_1 + I_2$$

$$\Delta O^2 = \sigma_1^2 + \sigma_2^2$$

Division:

$$O = I_1 \; / \; I_2$$

$$\Delta O^2 = \frac{\sigma_1^2}{I_2^2} + \frac{I_1^2 \sigma_2^2}{I_2^4}$$

Multiplication:

$$O = I_1 \; . \; I_2$$

$$\Delta O^2 = I_2^2 \sigma_1^2 + I_1^2 \sigma_2^2$$

Square-root:

$$O = \sqrt(I_1)$$

$$\Delta O^2 = \frac{\sigma_1^2}{I_1}$$

Logarithm:

$$O = log(I_1)$$

$$\Delta O^2 = \frac{\sigma_1^2}{I_1^2}$$

Polynomial Term:

$$O = I_1^n$$

$$\Delta O^2 = (nI_1^{n-1})^2\sigma_1^2$$

Square-root of Sum of Squares:

$$O = \sqrt{I_1^2 + I_2^2}$$

$$\Delta O^2 = \frac{I_1^2\sigma_1^2 + I_2^2\sigma_2^2}{I_1^2 + I_2^2}$$

Notice that some of these results are independent of the image data. Thus these algorithms preserve uniform random noise in the output image.

Such techniques form the basis of the most useful building blocks for image processing algorithms.

Some however, (most notably multiplication and division) produce a result which is data dependent, thus each output pixel will have different noise characteristics. This complicates the process of algorithmic design.

Complicated image processing algorithms are likely to have complicated derivatives (c.w. Bowyers Conjecture).

Linear Filters.

For Linear Filters we initially have to re-introduce the spatial subscript for the input and output images I and O.

$$O_{xy} \; = \; \sum_{nm} h_{nm} I_{x+n,y+m}$$

where h_{nm} are the linear co-efficients.

Error propagation gives:

$$\Delta O_{xy}^2 \; = \; \sum_{nm} (h_{nm} \sigma_{x+n,y+m})^2$$

for uniform errors this can be rewritten as

$$\Delta O_{xy}^2 \; = \; \sigma^2 \sum_{nm} (h_{nm})^2 \; = \; K \; \sigma^2$$

Thus linear filters produce outputs that have uniform errors.

Unlike image arithmetic, although the errors are unform they are no-longer independent because the same data is used in the calculation of the output image pixels. Thus care has to be taken when applying further processing.

For the case of applying a second linear filter this is not a problem as all sequences of linear filter operations can be replaced by a combined linear filter operation, thus the original derivation holds.

Histogram Equalisation.

For this algorithm we have a small problem as the differential of the processing process is not well defined.

If however we take the limiting case of the algorithm for a continuous signal then the output image can be defined as:

$$O_{xy} = \int_0^{I_{xy}} f dI / \int_0^{\infty} f dI$$

where f is the frequency distribution of the grey levels (ie the histogram.

This can now be differentiated giving

$$\frac{\partial O_{xy}}{\partial I_{xy}} = K \ f_{I_{xy}}$$

ie: the derivative is proportional to the frequency of occurrence of grey level value I_{xy} and the expected variance is:

$$\Delta O_{xy}^2 = K \ \sigma_{xy}^2 f_{I_{xy}}^2$$

Clearly this will not be uniform across the image, nor would it be in the quantized definition of the algorithm.

Thus although histogram equalisation is a popular process for displaying results (to make better use of the dynamic range available in the display) it should generally be avoided as part of a Machine Vision algorithm.

Monte-Carlo Techniques.

Differential propagation techniques are inappropriate when:

- Input errors are large compared to the range of linearity of the function.
- Input distribution is non-Gaussian.

The most general technique for algorithm analysis which is still applicable under these circumstances is known as the Monte-Carlo technique.

This techniques takes values from the expected input distribution and accumulates the statistical response of the output distribution.

The technique requires simply a method of generating random numbers from the expected input distribution and the algorithm itself.

Edge Detection.

Edge detection is a combination of operations and the simplest approach to testing is likely to be Monte-Carlo.

Canny was designed to combine optimal noise suppression with location accuracy, but does this account for its stability?

The sequence of processing involves;

- convolution with the noise filter
  (eg: $\otimes$ Gaussian)

- calculation of spatial derivatives
  (eg: $\otimes$ (-1, 0, 1))

- calculation of edge strength
  (eg: $\sqrt{(\nabla_x^2 + \nabla_y^2)}$)

- thresholding and peak finding

The final stage will be reliable provided that we have stability after the first three image processing steps.

Feature Detection Reliability.

Generally, when locating features, we are interested in a limited set of performance characteristics.

- Position and orientation accuracy
- Detection reliability
- False Detection rate

The first of these can be performed using a Monte-Carlo repeatability experiment.

The last two require a gold standard against which to make a comparison.

In addition, most feature detection algorithms have a sensitivity threshold (which corresponds to the probability level of the null hypothesis). The best value will be data dependent.

The way to deal with this is to produce curves which describe the detection and false detection rates as a function of threshold (ie: ROC as described in the earlier part of this tutorial).

Invalid Assumptions.

We have reviewed some of the more common assumptions made regarding the statistical characteristics of data, but there are many more.

For example in our own TINA machine vision system a series of processes are used to extract stereo data at features and then locate an object in the scene.

Algorithm	**Assumptions**
Edge Detection	Edges present as expected
2D Fitting	Curves and lines can be linked and fitted
Stereo Matching	Accurate camera geometry metrics
3D Geometry	Accurate camera calibration
Wireframe Matcher	Gaussian errors
3D Locator	Closed form appropriate
Sequential Model Builder	Accurate features

Algorithmic assmuptions for the original 3DMM

Closing the vision loop can improve on object location accuracy by eliminating invalid assumptions.

Demo: Closed Loop Validation. (Lacey et. al. 2001)

Conclusions.

Statistical assumptions underpinning algorithmic approaches should be understood and data tested to confirm that it conforms to the expected distributions.

The most effective/robust algorithms will be those that match most closely the statistical properties of the data.

An algorithm which takes correct account of all of the data will yield an optimal result (Lorusso et.al. 1995)

Robust approaches, which take account of outliers, are generally needed in any practical algorithm.

We can use statistical methods to estimate the errors on estimated quantities.

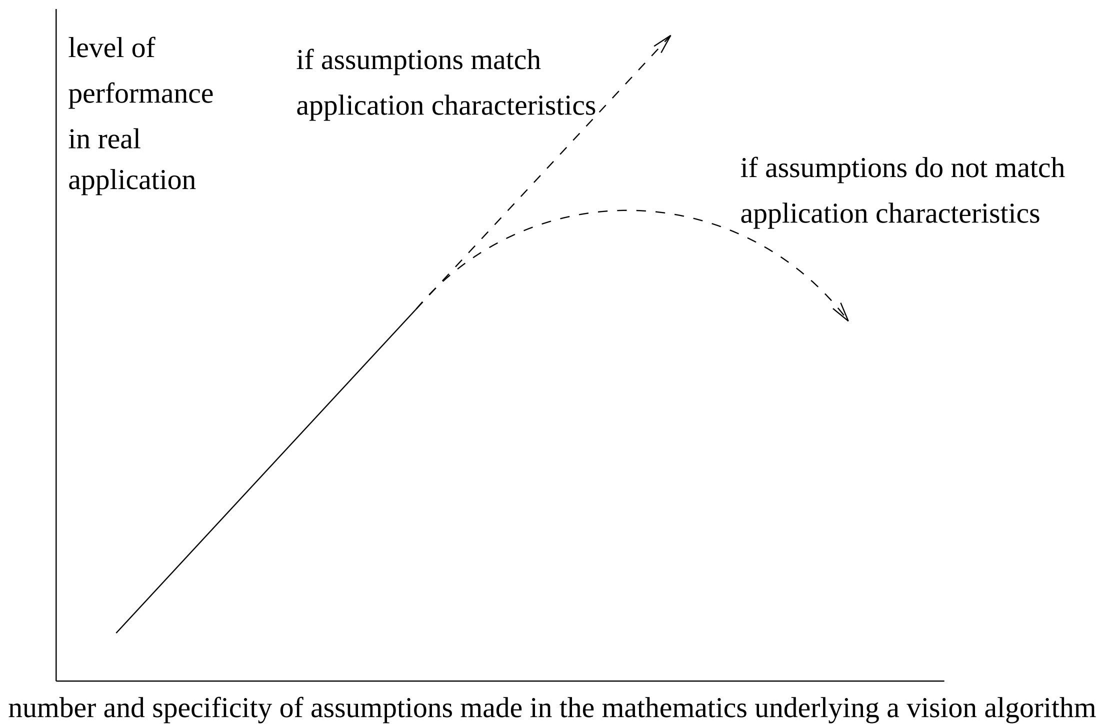
These errors are generally needed to make practical use of the data.

We can use methods such as error propagation to evaluate algorithmic functions analytically.

Monte-carlo methods can be applied where the analytic methods are inappropriate.

Complex functions generally have complex error characteristics, this is a theoretical justification for Bowyers Conjecture.

Algorithms which make many assumptions have more chance of one of them being invalid.



Algorithmic complexity should be increased only when it is justified by the data.

References.

1. R. M. Haralick, Covariance Propagation in Computer Vision, Proceedings of the ECCV Workshop on Performance Characteristics of Vision Algorithms, Cambridge, UK, April 1996.

2. Lorusso, A., Eggert, D.W., and Fisher, R.B., A Comparison of Four Algorithms for Estimating 3-D Rigid Transmformation, British Machine Vision Conference, Birmingham, UK, September 1995.

3. P.D. Sozou, T.F Cootes, C.J. Taylor and E.C. Di Mauro, Non-linear Point Distribution Modelling using a Multilayer Perceptron, British Machine Vision Conference Birmingham, UK, September 1995.

4. W.H.Press, B.P.Flannery, S.A.Teukolsky, W.T.Vetterling, Numerical Recipes in C, Cambridge University Press 1988.

5. A.P.Ashbrook, N.A.Thacker, P.I.Rockett and C.I.Brown, ‘Robust Recognition of Scaled Shapes Using Pairwise Geometric Histograms.’, proc, BMVC 95 Birmingham, 503-512, July 1995.

6. A.Lacey, N.A.Thacker, P.Courtney and S.Pollard. TINA 2001: The Closed Loop 3D Model Matcher. To be presented at BMVC 2001.

All software used in this tutorial is available from:

www.niac.man.ac.uk/Tina