Tina Memo No. 2002-005
First deliverable to BAe systems

An Empirical Design Methodology for the Construction of Computer Vision Systems

N. A. Thacker, A. J. Lacey and P. Courtney.

Last updated
14/05/2003

Imaging Science and Biomedical Engineering Division,
Medical School, University of Manchester,
Stopford Building, Oxford Road,
Manchester, M13 9PT.

1 Background

Our approach to the construction and evaluation of systems is based upon what could be regarded as a set of self evident propositions.

- Vision algorithms must deliver information allowing practical decisions regarding interpretation of an image.
- Probability is the only self-consistent computational framework for data analysis, and so must form the basis of all algorithmic analysis processes.
- The most effective and robust algorithms will be those that match most closely the statistical properties of the data.
- A statistically based algorithm which takes correct account of all available data will yield an optimal result[1].

Attempting to solve vision problems of any real complexity neccesitates, as in other engineering disciplines, a modular approach (a viewpoint popularised as a model for the human vision system by David Marr [14]). Therefore most algorithms published in the machine vision literature attend to only one small part of the "vision" problem, with the implicit intention that the algorithm could form part of a larger system[2]. Bringing these together as components in a system requires that the statistical characteristics of the data generated by one module match the assumptions underpinning the next.

In many practical situations problems cannot be easily formulated to correspond exactly to a particular computation. Compromises have to be made, generally in assumptions regarding the statistical form of the data to be processed, and it is the adequacy of these compromises which will ultimately determine the success or failure of a particular algorithm. Thus, understanding the assumptions and compromises of a particular algorithm is an essential part of the development process. The best algorithms not only model the underlying statistics of the measurement process but also propagate these effects through to the output. Only if this process is done correctly will algorithms form robust components in vision systems.

The evaluation of vision systems cannot be separated from the design process. Indeed it is important that the system is *designed for test* by adopting a methodology within which performance criteria can be adequately defined. When a modular strategy is adopted, system testing can be usefully considered as a two stage process [17] (summarised in Figure 1);

- the evaluation of the statistical distributions of the data and comparison with algorithmic assumptions in individual modules; **technology evaluation**,
- the evaluation of the suitability of the entire system for the solution of a particular type of task; **scenario evaluation**.

The process of scenerio evaluation is often time consuming and not reusable. The process of technology evaluation is complex and multi-dimensional, however the results are reusable for a range of applications. It therefore merits effort and should be attempted. Ideally, we

[1]Where the definition of optimal can be unambiguosly defined by the statistical specification of the problem.

[2]Though it could be argued that many researchers have lost focus on the bigger problem and thus the true motivations of a modular approach.

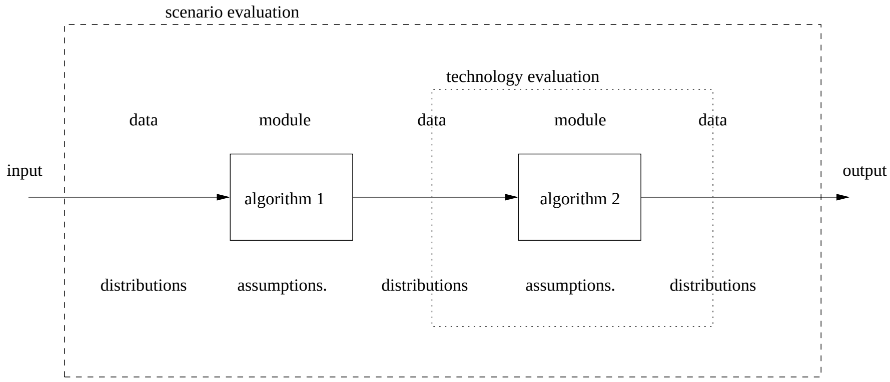


Figure 1: **Scenario** and **Technology** evaluation in a two stage statistical data analysis system.

would like to be able to specify a limited set of summary variables which define the requirements of the input data and the main characteristics of the output data, in a manner similar to an electronic component databook. However, it must be remembered that it is the suitability of the output data for use in later modules which defines performance, and in some circumstances it may not be easy (or even possible) to define performance independent of practical use of the data. For instance, problems can arise when the output data of one algorithm is to be fed into several subsequent algorithms, each having different or even conflicting requirements. The most extreme example of this is perhaps scene segmentation. A concise method for the evaluation of such algorithms in the absence of a definite goal is likely to continue to be a challenge [21]. Machine vision research has not emphasised the need or necessary methods of algorithm characterisation which is rather unfortunate, as the subject cannot advance without a sound empirical base [9]. In our opinion this problem can generally be attributed to one of two main factors; a poor understanding of the role of assumptions and statistics; and a lack of appreciation of what is to be done with the generated data. The assumptions behind many algorithms are rarely clearly stated and it is often left to the reader to infer them[3]. The failure to present clearly the assumptions of an algorithm often leaves the reader confused as to the novel or valid aspects of the published research and can give the impression that it is possible to create good algorithms by accident rather than design. In addition, the inability to match algorithms to tasks may lead those who require practical solutions to real problems to conclude that little (if anything) published in this area really works. When in fact, virtually all published algorithms can be expected to work, provided the input data satisfy the assumptions implicit in the technique. It is the unrealistic nature of these assumptions (e.g. noise free data) which is more likely to render algorithms useless. The following is a description of a methodology for the design of vision module components. This methodology focuses on identifying the statistical characteristics of the data and matching these to the assumptions of particular techniques. The methods given in the appendices have been drawn from over a decade of vision system design and testing, which has culminated in the construction of the TINA machine vision system [25]. These include a combination of standard techniques and less standard ones which we have developed to address specific problems in algorithm design.

[3] A process we have previously called "inverse statistical identification" an allusion to the analogous problem of system identification in control theory.

Example Task	**Data**	**Error Assumption**
Image Processing	Images	Uniform random Gaussian
Distribution Analysis	Histograms	Poisson sampling statistics
Shape Analysis	Edge location	Gaussian perpendicular to edge
2D Geometrical Location	Line fits	Uniform Gaussian on end-points
Object Tracking	Corner features	Circular (Elliptical) Gaussian
3D Objection Location	Stereo data	Gaussian in disparity space

Table 1: Standard error model assumptions.

2 Technology and Scenario Evaluation

There are several common models for statistical data analysis, all of which can be related at some stage to the principle of maximum likelihood (appendix A). This framework provides methods for the estimation and propagation of errors, which are essential for characterising data at all stages in a system. Likelihood based approaches begin by assuming that the data under analysis conforms to a particular distribution. This distribution is used to define the probability of the data given an assumed model (appendix B).

2.1 Input data

The first step in evaluating an algorithmic module is identification of the appropriate model and empirical confirmation of the distribution with sample data. Appropriate methods for this task include; correlation analysis, histogram fitting and the Kolmogrov-Smirnov test [24]. The interpretation of the results from such processes require knowledge of the consequences of deviation from the expected distribution. In general, the greatest problems are caused by outliers (see below) although, the better the data distributions conforms to the assumed model, the better the results. Assumptions which prove valid for one algorithm, can often prove useful in the design of new algorithms. Some distributions commonly used in the machine vision literature are tabulated in Table 1.

Although there are no general restrictions on the shape of these distributions the most common are Gaussian, Binomial, Multinomial and Poisson. These correspond to commonly occurring data generation processes and are well described in the statistical literature. The central limit effect ensures that the assumption of Gaussian distributed data forms the basis of many algorithms. This leads to tractable algorithms as the log-likelihood formulation of a Gaussian assumed model often takes the particularly simple form of a least-squares statistic, which can often be formulated as a closed form solution (appendix B). It is therefore useful to know that certain non-linear functions will transform the other common distributions to a form which approximates a Gaussian with sufficient accuracy to enable least-squares solutions to be employed.

Unfortunately, most practical situations generate data with long tailed distributions (outliers). The problems associated with outliers in data analysis are well known. However, what appears less well understood is the reason for the complete lack of closed form solutions based upon a long tailed distribution. Almost by definition only a simple quadratic form (or monotonic mapping thereof) for the log-likelihood, can have a unique minimum. Long tailed (non-Gaussian) likelihood distributions inevitably result in multiple local minima which can only be located by explicit search (e.g. the Hough transform) or optimisation (e.g. gradient descent).

Other assumptions in the likelihood formulation generally include those of data independence. Independence can be confirmed by plotting joint distributions. Uncorrelated data will produce joint distributions which are entirely predicted by the outer product of the marginal distributions (appendix F). Correlations (the lack of independence) in data can have several consequences. Strong correlations may produce suboptimal estimates from the algorithm and covariances may not concisely describe the error distribution (see later).

2.2 Output data

The next step in module analysis is to estimate the errors on the output data. If the output is the result of a log-likelihood measure then errors can be computed using covariance estimation (appendix C). Covariance estimation is possible even in the presence of outliers, provided that a robust kernal is used [15]. If the output quantities from a module are computed from noisy data the errors on the results can be calculated using error propagation (appendix D). Both of these theoretical techniques assume Gaussian distributed errors and locally linear behaviour of the algorithmic function.

These assumptions require validation (i.e. checks to ensure that the theory is an accurate representation of reality), which can be achieved using Monte-Carlo approaches (appendix H). Once again, techniques such as histogram fitting and Kolmogorov-Smirnov tests are useful. High degrees of non-linear behaviour can be addressed using a technique we call modal arithmetic [30] (appendix G). Non-linear transformation of estimated variables may be necessary in order to make better approximations to Gaussian distributions. It may also be necessary to combine variables in order to eliminate data correlation. The definition of the parameters passed between algorithms can be substantially different to naive expectation e.g. 3D data from a stereo algorithm is best represented in disparity space (appendix E). Selecting data representations which provide concise descriptions of statistical distributions is of fundamental importance [4]. Notice, the evaluation process is having direct influence on the process of system design, underscoring the earlier statements that system design and performance evaluation cannot (and should not) be treated separately.

In many cases the division of tasks into modules will be driven by the statistical characteristics of the processed data and cannot be specified *a priori*. Given the source of data typical of machine vision applications it is also very likely that algorithms will produce outlier data which cannot be eliminated by transformation or algorithmic improvement and will therefore require appropriate (robust) statistical treatment in later modules (see appendix L).

A rigid application of the above design and test process (see Figure 2) will produce verifiably optimal outputs from each module. Ultimately however, we will need to know if this data is of sufficient quality to achieve a particular task, a process we will call **scenario evaluation**. Under many circumstances it should be sufficient to determine the required accuracy of the output data in order to achieve this task. The covariance estimates from the technology evaluation would then be sufficient to quantify the expected performance of the system on a per-case basis.

Statistical measures of performance can be obtained by testing on a representative set of data. We would anticipate the need to compute the probability of a particular hypothe-

[4] yet is often overridden by preconcieved ideas of algorithm design.

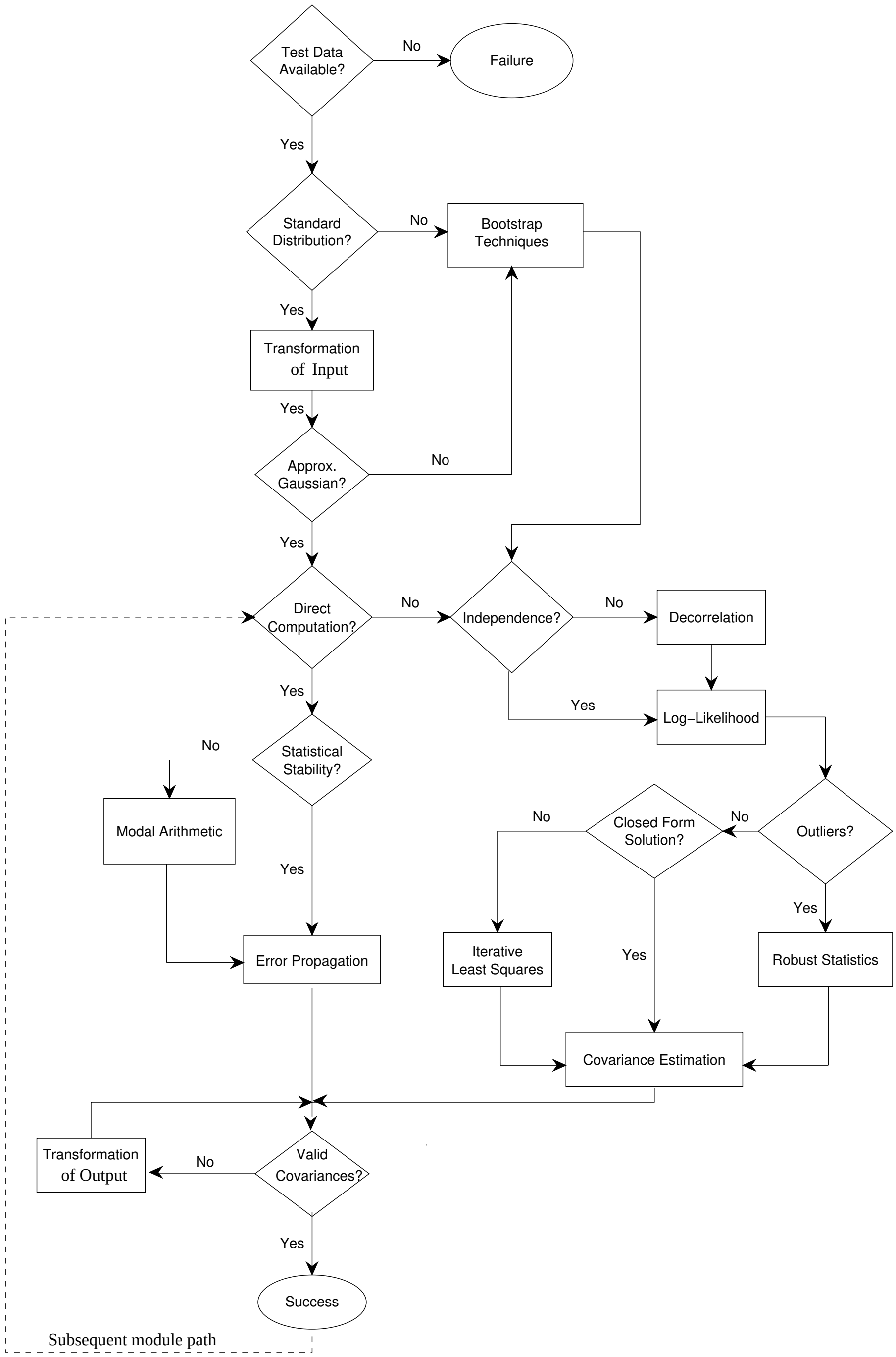


Figure 2: Technology evaluation flow chart. This diagram identifies the major design decisions which must be addressed in order to deliver quantified outputs from an algorithm. Transforms are suggested at various stages in order to solve problems associated with non-Gaussian behaviour. The label Bootstrap is intended to refer to coustom made statistical measures constructed from sample data.

sis, either as a direct interpretation of scene contents or as the likely outcome of an action (appendix I). Such probabilities are directly testable by virtue of being *honest probabilities* (appendix J). The term honest simply means the computed probability values should directly reflect the relative frequency of occurrence of the hypothesis in sample data (classification probabilities $P(C|data)$ should be wrong $1 - P(C|data)$ of the time). Tested hypothesis, such as that a particular set of data could have been generated by a particular model, should have a uniform probability distribution. Direct confirmation of such characteristics provide powerful tools for the quantitative validation of systems and provide mechanisms for online self test.

Often, we will need to construct systems which are not simply a series of sequential operations. It is quite likely that vision modules might provide evidence from several independant sources. Under these conditions we will need to perform a data fusion operation. Within the probabilistic framework described above there are three ways of achieving this; combination of probability (using a learning technique such as a neural network), combination of likelihoods (using covariances), and combination of hypothesis tests. All three of these are described in great detail in appendix K.

3 Identifying Candidate Modules from the Literature

Armed with the above methodology we are in a position to evaluate work in the machine vision literature in terms of its likely suitability for use in a vision system. In fact we can generate a short list of questions which exemplify those we should attempt to answer when evaluating a module for inclusion in a system.

- Does the paper make clear the task of the module?
- Are the assumptions used in algorithm design stated?
- Is the work based upon (or related to) a quantitative statistical framework?
- Are the assumed data distributions realistic i.e. representative of the data available in the considered situation?
- Has the computation of covariances been derived and tested?
- Does the theoretical prediction of performance match practical reality?
- Is the output data in a suitable form for use in any subsequent system?

The poor intersection between this list and general academic interests in this area (such as novelty and mathematical sophistication) underscores the main problems faced by those attempting to construct practical systems.

Notice that this list does not include system testing on typical image datasets, as that would be regarded as scenario rather than technology evaluation. Scenario evaluation, without considering the statistical characterisitcs of the data, is likely to be of much less value in the development of re-usable modules as the results will be task specific. Unfortunately, when performance characterisation is carried out in the literature it is very often a scenario evaluation. This goes against the implied assumption that most vision research is intended for use in a larger system.

4 Summary and Conclusions

This document suggests a quantitative statistical approach to the design and testing of machine vision systems which could be considered as an extension of methodologies suggested by other authors [2, 10]. We have focused on the use of likelihood and hypothesis testing paradigms and it would be natural for a reader familiar with the machine vision literature to feel that we have missed out other approaches which have (or have had) a higher profile in the literature (e.g. computational geometry and image analysis as inverse optics). However, we would argue that for the modular approach to system building to succeed we must have appropriate control over the statistical distributions generated during analysis. This is possible with likelihood based techniques because they enable the construction of measures to determine the best interpretation of the data (such as least squares) and also allow quantitative predictions to be made of the stability of estimated parameters (such as covariances). The machine vision problem, therefore, does not stop once a closed form solution is found (see [11] for a discussion of the use of statistics in closed form solutions). Inevitably, to acquire quantitative data for use in a systemi, error analysis will be required. This difficult step is often missing in the work found in the literature, yet attempting to do it can completely alter our understanding of the apparent value or even validity of the approach. The work of Maybank [13] demonstrated exactly this point with regard to the use of affine invariants for object recognition.

The reader may at this point feel that there is a broader context for probability theory than likelihoods and hypothesis testing. In particular likelihood based techniques have well known limitations, such as bias in finite samples [6]. The problem of model selection [27] is endemic in the machine vision area and likelihoods cannot be directly compared between two different model hypotheses. Approaches which aim to directly address these issues are thus acceptable extensions to the above methodology. However, some popular areas of probability theory do not (at least yet) have comparable quantitative capabilities (e.g. Bayesian approaches) and may therefore be unsuitable for system building. We have made an attempt to summarise these issues in [4]. It remains to be seen whether advocates of these approaches are able to address these issues.

Other approaches to algorithm design use methods which are based upon apparently different principles, such as entropy and mutual information [31]. However, we regard these as only alternative ways to formulate problems and believe that most experienced researchers would accept that all approaches should be reconcilable with probability theory [22]. Thus if there already exists a likelihood based formulation of the technique, this should be taken as the preferred approach. Obviously, if the research community as a whole accepted this viewpoint many papers would already have been written and presented differently. As the construction of systems from likelihood based formulations is generally likely to require optimisation of robust statistics, generic algorithms for the location of multiple local optima should be regarded as a fundamental research issue. So too should the problem of covariance estimation from common optimisation tasks and popular algorithmic constructs, (such as Hough transforms), which have already been shown to be consistent with likelihood approaches [23].

Many attempts at algorithmic evaluation in the literature focus on the specification of particular performance metrics. Although these metrics may give some indication as to the basic workings of an algorithm, quantitative evaluation should set as the ultimate goal an understanding of the performance of the system. Performance metrics for modules should therefore be specified with this in mind. Non-quantitative evaluation is probably of more use in the early stages of algorithm construction than during the final integration into a system.

In the methodology described a key aspect is the identification of assumptions. Knowledge of these assumptions (and suitable methods for determining their validity) allows comparisons of algorithms to be carried out at the theoretical level. Also, we should not be surprised when algorithms which are built upon the same set of founding assumptions within a sensible probabilistic framework, give near identical performance. This has been well illustrated in one particular piece of work by Fisher et. al [8], where alternative techniques for location of 3D models in 3D range data were found to give equivalent results to within floating point accuracy. If careful statistical analysis of data did not give this result then it would be an indication that probability theory itself was not self-consistent. Also, when performing comparative testing of modules we should be aware that algorithmic scope, as determined by the restrictions imposed by the assumptions, should be taken into account in the final interpretation of results. Algorithms which give apparently weaker performance on the basis of performance metrics may still be more applicable for some tasks. A simple example of this is that least squares fitting will generally give a better bounded estimate of a set of parameters than robust techniques, yet robust techniques are essential in the presence of outliers. An evaluation of these two techniques in the absence of outliers would incorrectly conclude that least-squares was always more accurate. Clearly this result is of limited use when building practical systems.

Appendix	Page	Title
A		Maximum Likelihood
B		Common Likelihood Formulations
C		Covariance Estimation
D		Error Propagation
E		Transforms to Equal Variance
F		Correlation and Independence
G		Modal Arithmetic
H		Monte-Carlo Techniques
I		Hypothesis Testing
J		Honest Probabilities
K		Data Fusion
L		Reciever Operator Curves

A Maximum Likelihood

A more detailed treatment of the theory and techniques of Maximum Likelihood statistics can be found in [6]. A summary of the theory is presented here for completeness.

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

$$P(X_0X_1X_2...X_n|Y)P(Y) = P(X_0|X_1X_2...X_nY)P(X_1|X_2...X_nY)......P(X_n|Y)P(Y)$$

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. For large numbers of variables this is an impractical method for probability estimation. 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

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

The rather redundant use of the conditional terms $|Y$ is often dropped for convenience. Clearly this is a more practical definition of joint probability but the requirement of independence is quite a severe restriction. However, in some cases data can be analysed to remove these correlations, in particular the use of an appropriate data model (such as in least squares fitting) and processes for data orthogonalisation (including principle component analysis). For these reasons all common forms of maximum likelihood definitions assume data independence.

Probability independence is such an important concept it is worth defining carefully. If knowledge of the probability of one variable A allows us to gain knowledge about another event B then these variables are **not** independent. Put in a way which is easily visualised, if the distribution of $P(B|A)$ over all possible values of B is constant for all A then the two variables are independent. Assumptions of independence of data can be tested graphically by plotting $P(A)$ against $P(B)$ or A against B if the variables are directly monotonically related to their respective probabilities.

There are two other aspects of the application which require particular care. The first is that we have described here probabilities of unique events. In fact for many applications of likelihood we will need to work with continuous functions. Alternative selections for the continuous data space will produce different likelihood estimates. This issue will be discussed further below. The second issue is related to the first and is the problem of probability normalisation. The standard definition simply demands that distributions be normalised to 1. Unfortunately, such normalisation does not take into account the true information content of the data. ML is therefore not a true statistic, one consequence of this is that distribution normalisation cannot be allowed to change during parameter estimation or ML estimates compared across data sets. However, estimation of quantitative statistical quantities, such as covariances (Appendix C) can proceed provided the normalisation is fixed.

B Common Likelihood Formulations

Dealing with Binary Evidence

The simplest likelihood model is for binary observations of a set of variables with known probabilities. 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:

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

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 minimised 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.

If we now average many binary measures of X into the vector O we can compute the mean probability of observing the distribution O generated from $P(X)$ as:

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

It should be noted that this is not the log probability that O is the same distribution as P as it is asymmetric under interchange of O and P. To form this probability we would also have to test for P being drawn from the distribution O. The resulting form of this comparison metric is often referred to as the log entropy measure as the mathematical form is analogous to some parts of statistical mechanics in physics.

Poisson and Gaussian Data Distributions

A very common problem in machine vision is that of determining a set of parameters in a model. Take for example a set of data described by the function $f(a, Y_i)$ where a defines the set of free parameters defining f and Y_i is the generating data set. If we now define the variation of the observed measurements X_i about the generating function with some random error we can see that the probability $P(X_0|X_1X_2...X_N aY_0)$ will be equivalent to $P(X_0|aY_0)$ as the model and generation point completely define all but the random error. Choosing Gaussian random errors with a standard deviation of σ_i gives:

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

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

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

which leads to the χ^2 definition of log likelihood

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

This expression can be maximised as a function of the parameters a and this process is generally called a least squares fit. Whenever you encounter least squares there is implicit assumption of independence and of a Gaussian distribution. In practical situations the validity of these assumptions should be checked by plotting the distribution of $X_i - f(a, Y_i)$ to make sure that it is Gaussian.

Often when working with measured data we need to interpret frequency distributions of continuous variables, for example in the form of frequency histograms. In order to do this we must know the statistical behaviour of these measured quantities. The generation process for a histogram bin quantity (making an entry at random according to a fixed probability) is strictly a multi-nomial distribution, however for large numbers of data bins this rapidly becomes well described by the Poisson distribution. The probability of observing a particular number of enties h_i for an expected probability of p_i is given by:

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

For large expected numbers of entries this distribution approximates a Gaussian with $\sigma = \sqrt{h_i}$. The limit of a frequency distribution for an infinite number of samples and bins of infinitessimal 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

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

This equation has the restriction that it is not defined in the region where $h_i + j_i = 0$. We can overcome this problem by transforming the data to a domain where the errors are uniform by taking square roots. This process not only reduces the approximation error but also removes the denominator, leading to the common form of probability comparison metric known as the Matusita distance measure L_M:

$$L_M = \sum_i (\sqrt{P_1(X_i)} - \sqrt{P_2(X_i)})^2$$

This can be rewritten in a second form:

$$= 2 - 2 \sum_i \sqrt{P_1(X_i) P_2(X_i)}$$

Where the second term defines the Bhattacharyya distance metric L_B:

$$L_B = \sum_i \sqrt{P_1(X_i)} \sqrt{P_2(X_i)}$$

C Covariance Estimation

The concept of error covariance is very important in statistics as it allows us to model linear correlations between parameters. 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 where the quadratic formula is:

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

This can be re-written in matrix algebra as:

$$\chi^2 = \chi_0^2 + \Delta X^T C_x^{-1} \Delta X$$

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

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

Comparing this with the original quadratic equation gives

$$\chi^2 = \chi_0^2 + \Delta X^2 u + \Delta Y \Delta X w + \Delta X \Delta Y v + \Delta Y^2$$

where

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

Notice that the b and c coefficients are zero as required if the χ^2 is at the minimum. In the general case we need a method for determining the covariance matrix for model fits with an arbitrary number of parameters. 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:

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

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

The second term in this equation is expected to be negligible compared to the first and with an expected value of zero if the model is a good fit. Thus the cross derivatives can be approximated to a good accuracy by

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

The following quantities are often defined

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

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

$$C = \alpha^{-1} \quad \text{where } \alpha = \begin{vmatrix} \alpha_{11} & \alpha_{12} & \dots \\ \alpha_{21} & \alpha_{22} & \dots \\ \dots & \dots & \alpha_{nm} \end{vmatrix}$$

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

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

D 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 the precision of this system. Therefore, we require a method for propagating likely errors on measurements through to $f(X)$. Assuming knowledge of error covariance this can be done as follows:

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

The method simply uses the derivative of the function f as a linear approximation to approximation to that function. This is sufficient provided that the expected variation in parameters ΔX is small compared to the range of linearity of the function. Application of this technique to even simple image processing functions gives useful information regarding the expected stability of each method (Table 2). When constructing algorithms from such image processing modules any data dependency will produce problems with noise stability unless the errors are propagated fully for later use.

Process	Calculation	Theoretical Error
Addition	$O = I_1 + I_2$	$\Delta O^2 = \sigma_1^2 + \sigma_2^2$
Division	$O = \frac{I_1}{I_2}$	$\Delta O^2 = \frac{\sigma_1^2}{I_2^2} + \frac{I_1^2\sigma_2^2}{I_2^4}$
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	$O = I_1^n$	$\Delta O^2 = (nI_1^{n-1})^2\sigma_1^2$

Table 2: Table 2: Error Propagation in Image Processing Operations

As an example we can take the Poisson distribution itself which for large numbers is expected to have a standard deviation of $\sqrt{N}$ where N is the mean of the distribution. We will call a sample random variable from this distribution s. If we now construct a new measure given by

$$t = \sqrt{s}$$

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

$$\begin{aligned}\Delta t &= \frac{\partial t}{\partial s}\Delta s \\ &= \frac{-1}{2\sqrt{s}}\sqrt{s} \\ &= \frac{-1}{2}\end{aligned}$$

Thus the distribution of the square-root of a random variable drawn from a Poisson distribution with large mean will be constant. This result can be used to generate the Matusita probability distribution comparison metric defined in appendix B.

When the problem does not permit algebraic mainpulation in this form (due to significant non-linear behaviour in the range of $\Delta f(X)$ or functional discontinuities) then numerical (Monte-Carlo) approaches may be helpful in obtaining the required estimates of precision (appendix H).

E Transforms to Equal Variance

The choice of a least squares error metric gives many advantages in terms of computational simplicity and is also used extensively for definitions of error covariance and optimal combination of data (Appendices C and K). However, the distribution of random variation on the observed data X is something that generally we have no initial control over and could well be arbitrary and so we have the problem of adjusting the measurements in order to account for this. In addition, we have the problem that different choices for the way we represent continuous data will produce different likelihood measures. Take for example a set of measurements made from a circle, we can choose to measure the size of a circle as a radius or as an area. However, it can be easily shown that constructing a likelihood technique based upon sampled distributions will produce different (inconsistent) formulations for these two representations of the same underlying data. Transferrring the likelihood from a distribution of radial errors will not produce the impirically observed distribution for area due to the non-linear transformation between these variables. Which should we choose as correct (or are both wrong)? Initially these may be seen as separate problems, but in fact they are related and may have one common solution. To understand this we need to consider non-linear data transformations and the reasons for applying them.

In many circumstances it is possible to make distributions more suitable for use of standard ML formulations (Appendices B) by transformation $g(X_i)$ and $g(f(a, Y_i))$, where g is chosen so that the initial distribution of X_i maps to an equal variance distribution (near Gaussian) in g. Examples of this for statistical distributions are the use of the square-root transform for Poisson distributed variables [28] and the $asin$ mapping for binomial distributed data. However, this problem can occur more generally due to the need to have to work with quantities which are not measured directly, but computed from other measured quantities.

One good example of this is in the location of a known object in 3D data derived from a stereo vision system. In the coordinate system where the viewing direction corresponds to the z axis, x and y measures have errors determined by image plane measurement. However, the depth z_i for a given point is given by

$$z_i = fI/(X_{li} - X_{ri})$$

where I is the interocular separation, f is the focal length and X_{li} and X_{ri} are image plane measurements. Attempts to perform a least squares fit directly in (x, y, z) space results in instability due to the non-Gaussian nature of the z_i distribution. However, transformation to $(x, y, 1/\sqrt{2}z)$ yields Gaussian distributions and good results. In general, observation of a dependency of the error distribution of a derived variable with that variable (in the above case the dependency of σ_z on z), is very often a sign that the likelihood distribution is skewed. For a known functional dependency h the transformation g which maps the variable X_i to one with equal variance follows directly from the method of error propagation and is given by

$$g \ = \ \int \frac{1}{h(X)} dX$$

All of the transformations mentioned above can be generated from this process, including those which map standard statistical distributions to more Gaussian ones, though the extent to which this is a general property of this method is unclear. We are now also in a position to answer our questions regarding data representation in ML. The selection of measured variables from the equal variance domain provides a unique solution to the problem of identification of the source data space.

F Correlation and Independence

Under practical circumstances the data delivered to an algorithm may be correlated. Generally, it is the job of the model used in the formulation of the likelihood approach to account for all expected systematic correlation up to a random independent noise process. However, likelihood formulations often assume data independence for simplicity. Correlation produces systematic changes in the residuals of 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:

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

that is, decomposable. We may wish to preprocess the data to remove these correlations using **Principal Component Analysis** in order to conform to the assumption of independence. We can define the correlation matrix:

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

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. So the solution of the eigenvector equation:

$$RV_k = \lambda_k V_k$$

defines the axes of a co-ordinate system V_k which decorrelates the data. The method known as Singular Value Decomposition (SVD) [19] approximates a matrix by a set of orthogonal vectors, W_l, and singular values, w_l and it can be shown that the singular vectors satisfy the eigenvector equation with:

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

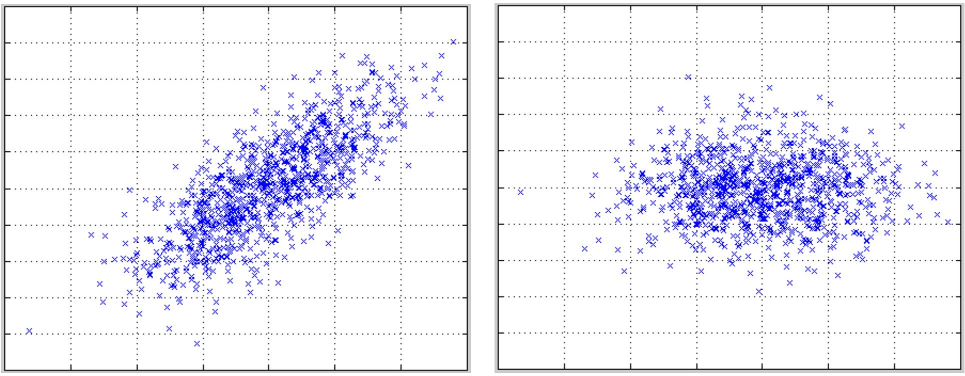
If we multiply both sides of the equation by one of these vectors W_k

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

we see that the singular vectors satisfy the eigenvector equation with

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

Thus, SVD determines the axes of maximal variation within the data. Figure 3(a) shows a scatter plot of original data, where can be seen that data are correlated. Figure 3(b) shows rotated data set using determined eigenvectors, hence data are uncorrelated in that new coordinate system, where eigenvectors are new coordinate axes.



(a) original (b) rotated

Figure 3: Original and rotated data distributions

A limited approximation to the full matrix R^*

$$R^* = \sum_{l}^{l_{max}} \frac{1}{{w_l}^2} W_l \otimes W_l$$

gives an optimal approximation to the matrix R in the least squares sense $(R - R^*)^2$, allowing the selection of a reduced number of orthogonal descriptor variables. This process of principle component analysis is useful for limiting the effects of numerical stability and singularity during the process of matrix inversion. The equivalence of eigenvectors and principle axis means that the eigenvectors align with the axis of maximum variance. This leaves a minimum variance on the excluded eigen vector directions, effectively a hyperplane fit.

G Modal Arithmetic

Sometimes the effects of non-linear calculations on data with a noise distribution affects not only the variance of the computed quantity but also the mean value. From a likelihood point of view we can define the ideal result from a computation as the most frequent (or modal) value that would have resulted from data drawn from the expected noise distribution. We have termed the algorithm design technique which addressed this issue *modal arithmetic*. The general method of modal arithmetic for a measured value with distribution $D(x)$ and a non-linear function $f(x)$ would be to find the solution x_{max} of

$$\partial [\frac{D(x)}{\partial f(x)/\partial x}]/\partial x \ = \ 0$$

with the modal solution of $f(x_{max})$.

Modal arithmetic is unconditionally stable, as peaks in probability distributions cannot occur at infinity. It also has much similarity with some approaches in statistics which advocate the use of the **mode** rather than the **mean** as the most robust indicator of a distributed variable. The simplest example of this is for image division (e.g. $y = 1/x$) where small errors on the data produce instabilities in computations involving large quantities of data. Error propagation shows that a small change in the input quantity Δ_x will give an error on the corresponding output of

$$\Delta_y \ = \ \frac{\Delta_x}{x^2}$$

which is clearly unstable for values of x which are comparable to its error. This problem can be understood better by considering the distribution of computed values from the range of those available for input. We start by assuming a Gaussian distribution for the denominator

$$P_x \Delta_x \ = \ A \ exp(-(x \ - x_0)^2 / 2\sigma^2)$$

Where x_0 is the central value of x with a standard deviation of σ.

If we take a small area of data from the probability distribution for x (i.e.: $P_x\Delta_x$), we can associate this with an equal number of solutions in the output space y (ie: $P_y\Delta_y$) (Figure 4 (a) and (b)) giving:

$$P_y \ = \ A \ x^2 exp(-(x \ - x_0)^2 / 2\sigma^2)$$

This expected probability distribution for y as a function of x (Figure 4(c)) can be differentiated to find its maxima.

$$\partial P_y / \partial x \ = \ 2A \ exp(-(x \ - x_0)^2 / 2\sigma^2)(x \ - \ x^2(x \ - \ x_0)/2\sigma^2)$$

Setting this to zero we can determine the modal values of this distribution:

$$x^2 - x_0 x - 2\sigma^2 \ = \ 0 \qquad with \qquad x_{max} \ = \ \frac{x_0 \pm \sqrt{x_0^2 \ + \ 8\sigma^2}}{2}$$

which correspond to the positive and negative peaks due to the distribution of x spanning zero (Figure 4(a)). If we were to ask which value of y would be most likely to result from the division then the answer would be $1/x_{max}$ selected with the same sign as the input value

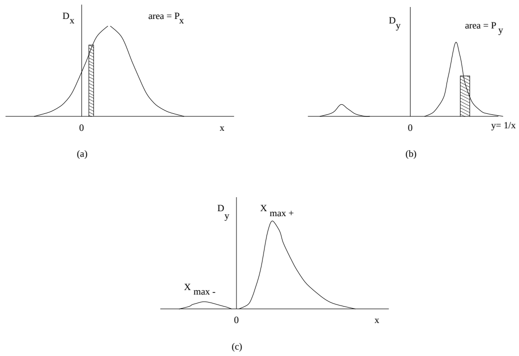


Figure 4: Probability Distributions for a noisy denominator.

x_0. Taking this value as a replacement for the denominator provides a maximum likelihood technique of stabilising the process of division using knowledge of measurement accuracy and could best be described as **modal division**. Modal division can be used with impunity for calculations involving large quantities of noisy data without instability problems for values around zero, with the minimum denominator limited to a value of $\sqrt{2}\sigma$. In previous work we were able to show that the application of modal arithmetic to image deconvolution regenerated the results from the standard likelihood based technique of Weiner filtering [32].

H Monte-Carlo Techniques

The concept of Monte-Carlo techniques is very simple. A computer simulation is performed which generates multiple sets of data from within the expected measurement distribution. These data are then passed through the algorithmic computation and the distributions of resulting values around their true values accumulated. This way both the systematic errors (bias) and statistical errors (variance) associated with the algorithm can be assessed. This is done either by comparing these distributions with results from covariance estmation or error propogation or by empirical construction of the dependency of the computed values on the input quantities [29]. These models can then be used to quantify the expected error distributions on the data when provided as input to other modules.

Key to the success of these techniques is the ability to generate random example of data drawn from the required distributions. We start by using a random number $0 < x < 1$ drawn from a uniform (flat) distribution. The general technique for generating a variate from the distribution $f(y)$ using x is to solve for y in

$$x = \int_{-\infty}^{y_0} f(y)dy / \int_{-\infty}^{\infty} f(y)dy$$

i.e. x is used to locate a variable some fraction of the way through the integrated distribution.

For instance, a Gaussian distribution leads to the BOX MULLER method [19];

$$y_1 = \sqrt{(-2ln(x_1))}cos(2\pi x_2)$$

$$y_2 = \sqrt{(-2ln(x_1))}sin(2\pi x_2)$$

which generates two Gaussian random deviates y_1 and y_2 for every two input deviates x_1 and x_2.

Armed with distribution generators we can generate many alternative images for statistical testing from only a few examples of image data.

I Hypothesis Testing

Having made quantitative measurements from our system we will ultimately need to make decisions based upon those measurements in comparison to some predefined model. For example, do not attempt to move the mobile vehicle through a doorway unless the vision system estimates that it will pass. Many statistical tests are based on the idea of generating the probability that data drawn from the expected test distribution would be more frequent than the example under test. This approach leads to the common statistical techniques of z-scores, T tests, and Chi-squared tests to name a few. This follows directly from the original definition of a confidence interval, due to Neyman [16] and yet is rarely used in machine vision. This is unfortunate, as the methods do not suffer the same restrictions regarding distributions which apply to covariances.

Hypothesis tests (i.e. does the data conform to the assumed model?) are perfomed on the basis of one model at a time, in contrast to Bayesian approaches which require all possible generators (models) of the data. In addition, such statistical tests are fully quantitative. Probabilities computed from such statistics have the characteristic that the distribution of values drawn from the assumed model will be flat. This is useful as a mechanism for self test. The most common form of this statistic is that for a Gaussian and is known as the **error function** which is provided as a mathematical function in most languages (e.g. the erf() library function). However, such statistics can be generated for any model for which the expected data distribution is known, using the ordering principle. This states that the ordering of integration along the measurement axis should be defined so that the probability density is monotonically decreasing. For the Gaussian case shown above this gives the rather trivial result that we integrate along the standard measurement axis x away from the peak, as the function is monotonically decreasing from $x = 0$. Although this is not the only way to order the data (there are potentialy infinite numbers of equivalent possible ordering schemes depending upon how we define our variables e.g. x^2) this is the one which gives confidence limits which are maximally compact in the chosen parameter domain. Generally, the preferred parameter domain would be selected as the space in which x was uniformly accurate, so that this compactness has meaning from the point of view of measureable localisation. This is sometimes referred to as a "natural" parameterisation and is related to the concept of the equal variance transform (appendix E).

In image processing the required distributions can often be bootstrapped directly from the image (e.g. as in [5]). Under these circumstances the possibility of multi-modal density functions makes the application of the ordering principle slightly less straightforward.

Finally, as the only requirement for the use of such probabilities is that they have a uniform distribution, empirical approaches can be used to re-flatten distributions which result from imprecise analysis. Such hypothesis tests are also easily combined using standard statistical approaches (See appendix K).

J Honest Probabilities

The correct use of statistics in algorithm design should be fully quantitative. Probabilities should correspond to a genuine prediction of data frequency. From the point of view of algorithmic evaluation, if an algorithm does not make quantitative predictions then it is by definition untestable in any meaningful manner. Thus a classifier which giving a probability of a particular class as P should be wrong $1 - P$ of the time. Probabilities with these characteristics have previously been referred to in the literature as honest [7]. The importance of this feature in relation to the work presented here is that knowledge of the expected distribution for the output provides a mechanism for self-test. For example classifier error rates can be assessed as a function of probability to confirm the expected correlation. Some approaches to pattern recognition, such as k-nearest neighbours, are almost guaranteed to be honest by construction. In addition the concept of honesty provides a very powerful way of assessing the validity of probabilistic approaches [18].

Supervised classification performance, for example object recognition, can be specified in terms of the confusion matrix. This is table that describes the probabilities that an item of class i will be misclassified as an item of class j for each of a set of classes. The sum of each of the rows and columns should add up to 1.0.

	class 1	class 2	class 3	class 4
class 1	1.0	0.0	0.0	0.0
class 2	0.0	0.8	0.15	0.05
class 3	0.0	0.15	0.35	0.5
class 4	0.0	0.05	0.5	0.45

Table 1: Confusion Matrix

A perfect classifier would have value of 1.0 along the diagonal where $i = j$ and zero elsewhere. However, a real classifier would have some off-diagonal elements, as in this example. Note that the table is not necessarily symmetrical. The classification algorithm might also specify a rejection rate at which it will refuse to produce a valid class output. For the problilities delivered by a classification system to be honest, the mean probability generated for each position in the confusion matrix should agree with the relative frequency of the sampled data. For example in the table given above class 1 should always be identified with 100% classification probability. This is rather a strong claim for any empirical estimate of performance and should at least be accompanied by the number of samples in the test set [5].

A technology evaluation would provide an unweighted table, but a scenario evaluation would weight the entries to take account of the prior probabilities of the various objects, according to a particular application and the cost of various types of error, to produce an overall number for ranking.

[5] Tests with N samples should restrict themselves to claims of only (1 - 1/N).

K Data Fusion

Optimal Combination using Covariances

Given two estimates of a set of parameters a_1 and a_2 and their covariances (α_1 and α_2) we can combine the two sets of data as follows:

$$a_T = \alpha_T^{-1}(\alpha_1 a_1 + \alpha_2 a_2) \quad with \quad \alpha_T^{-1} = \alpha_1^{-1} + \alpha_2^{-1}$$

This method combines the data in the least squares sense, that is the approximation to the χ^2 stored in the covariance matrices has been combined directly to give the minimum of the quadratic form. The method can be rewritten slightly giving

$$a_T = a_1 + \alpha_T^{-1}\alpha_2 \Delta a$$

where $\Delta a = a_2 - a_1$. This form is directly comparable to the information filter form of the Kalman filter.

Optimal Combination of Hypothesis Tests

Hypothesis test probabilities should have uniform distributions (if they are honest). Given n quantities each having a uniform probability distribution $p_{i=1,n}$, the product $p = \prod_{i=1}^{n} p_i$ can be renormalised to have a uniform probability distribution $F_n(p)$ using:

$$F_n(p) = p \sum_{i=0}^{n-1} \frac{(-\ln p)^i}{i!} \tag{1}$$

Proof of this relationship can be generated in the following manner. The quantities p_i can be plotted on the axes of an n dimensional sample space, bounded by the unit hypercube. Since they are uniform, and assuming no spatial correlation, the sample space will be uniformly populated. Therefore, the transformation to $F_n(p)$ such that this quantity has a uniform probability distribution can be achieved using the probability integral transform, replacing any point in the sample space p with the integral of the volume under the contour of constant p. Generalisation of this process to non-integer numbers (which is useful for cases where we have an effective number of degrees of freedom) and other useful results are presented in [3].

Optimal Combination from Example Data

When the area of neural networks re-emerged as a popular topic in the mid 80's much was claimed about the expected capabilities regarding flexibility, suitability for system identification and robustness. Most of these claims were subsequently shown to be optimistic. However, one problem that neural networks are relatively good at is non-linear data fusion. A neural network when trained on an appropriate form of data with the correct algorithm will approximate Bayes probabilities as outputs.

The mathematics describing this process is given in [12] but a more intuitive argument is as follows. Each input vector pattern X defines a unique point in input space. Associated with each data point is the ideal required output, for example a binary output classification. As the number of samples grows large the number of examples of data in the region of each point also grows large. If training with a least squares error function the target output for each point in pattern space will be the mean of local values. For a binary coding problem the mean value is the Bayes probability of the model given the data.

Given $P(A|B)$ and $P(A|C)$ can we compute $P(A|BC)$? We can clearly solve this problem provided these probabilities are independent by simple multiplication. If however the measures are correlated there is no standard statistical method for this process. This is unfortunate as we would expect a modular (AI) decision system to need to solve this task. Standard neural network architectures trained in the standard way will however approximate $P(A|P(A|B)P(A|C))$ for the reasons described above [1]. Provided that there is enough information in the set of probabilities being fused to regenerate the original data the fusion process will be able to achieve optimality.

L Reciever Operator Curves

Scenario evaluations in machine vision often result in the problem of establishing how well an algorithm can identify a particular situation in the image. The simplest example of this problem is a feature detector. Feature detection reliability has two elements: the probability of the detection of a true feature (True Acceptance Rate or TAR), and the probability of the detector signalling a feature which is in fact absent (False Acceptance Rate or FAR). These may be represented as two probability density functions (pdf): the signal and the non-signal pdfs.

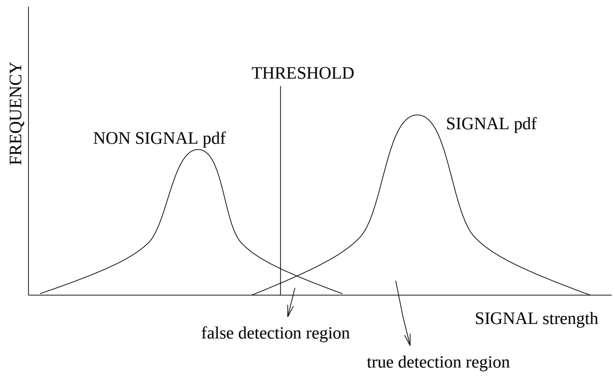


Figure 5: signal and non-signal detection pdfs

A feature detector generally has a threshold which allows a trade off to be made between the two types of error. For a given threshold the true acceptance rate will be the area under the curve of the signal pdf and to the right of the threshold, whereas the false acceptance rate is the area under the curve of the non-signal pdf and to the right of the threshold. This gives rise to two extreme situations. If the threshold is set to the far left, the detector will accept all the signal but also all non-signal, so both TAR and FAR will be high. If the threshold is set to the far right, the detector will reject all non-signals, but also reject all true signals, so both TAR and FAR will be low. It is important to appreciate that for detection algorithms there is always a trade off between true and false detection.

An understanding the behaviour of a feature detection algorithm as the threshold is varied can be obtained by plotting an ROC (receiver operating characteristic) curve.

In the ROC curve [6], one axis represents the True Acceptance Rate (TAR) and the other represents the False Acceptance Rate (FAR) [7]. Each runs from 0% to 100%. The performance of a given detection algorithm may be described in terms of a line passing through various combinations of TAR and FAR. The ideal algorithm would be one with a line that passes as close as possible to the point TAR=100% and FAR=0%. The precise location along the line is determined by the setting of the threshold parameter described earlier. The setting of the threshold is made on the basis of the consequence of each type of error (Bayes risk), and this will depend on the use of the results and thus the application, subject to prior probabilites of the signal and non-signal.

The performance of detection algorithms is sometimes quoted in terms of the equal error

[6]note that there appear to be no conventions as to the orientation of the plot

[7]a number of alternative forms are used such as reject rate which is (1 - acceptance rate)

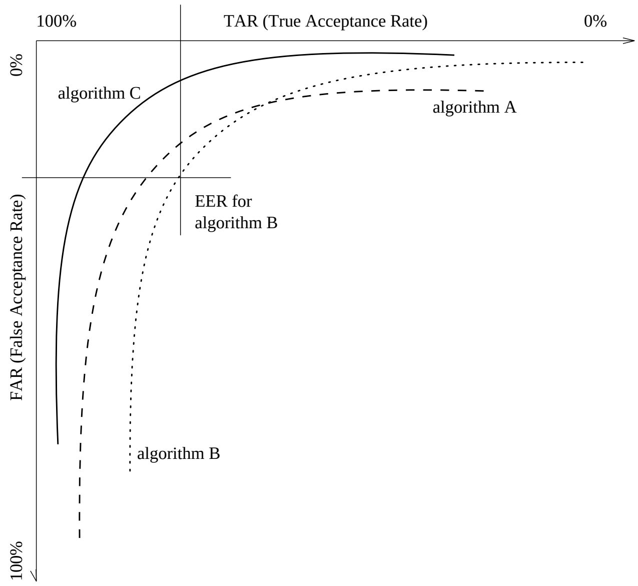


Figure 6: Receiver Operator Curve

rate (EER). This is the point at which the FAR is equal to the True Reject Rate (TRR=1-TAR). This may be appropriate for some applications in which the cost of each type of error is equal. However this is not generally the case so access to the entire ROC curve is preferred.

In contrast to the earlier pdf diagram, the performance of different algorithms may be presented on the same plot and thus compared. For a given application (and thus TAR/FAR trade off) one algorithm may be superior to another according to the desired position along the ROC curve. For instance algorithm B may be superior to algorithm A when a low FAR is required. Conversely algorithm A will be preferred when a high TAR is required. Algorithm C on the other hand provides superior performance to both algorithm A and algorithm B since for each value of FAR, algorithm C will have a higher level of TAR.

Notice the difference between the ROC plot that *presents* the performance characteristics of a number of algorithms (the result of a technology evaluation), and the decision as to which is the best and how it should be tuned, which is based on the *use* of this information (scenario evaluation). Of course the ROC curve is only as good as the data used to generate it, and a curve produced using unrepresentative data will not be applicable to the task.

There are variants of the ROC curve. If the task is to identify the features in an image, and it is possible that there will be more than one, then a fractional ROC (FROC) is more appropriate. This plots the total number of false detection (since there may be more than one) against the probability of a true detection as before.

The fact that every detection algorithm involves a trade off between true and false detections has the consequence that false detections must **be tolerated** by the subsequent processing stages.

References

1. D.BOOTH, N.A.THACKER, M.K.PIDOCK AND J.E.W.MAYHEW. *Combining the Opinions of Several Early Vision Modules Using a Multi-Layered Perceptron.* Int.Journal of Neural Networks, 2, 2/3/4, June-December, 75-79, 1991.

2. K. W. BOWYER AND P. J. PHILLIPS *Empirical Evaluation Techniques in Computer Vision* edited by K.W. Bowyer and P.J. Phillips, IEEE press, ISBN 0-8186-8401-1, 2000

3. P. A. BROMILEY, T.F. COOTES AND N.A. THACKER, *Derivation of the Renormalisation Formula for the Product of Uniform Probability Distributions and Extension to Non-Integer Dimensionality.* Tina memo 2001-008.

4. P.A. BROMILEY, M.L.J. SCOTT, M. POKRIĆ, A.J. LACEY AND N.A. THACKER, *Bayesian and Non-Bayesian Probabilistic Models for Magnetic Resonance Image Analysis*, Submitted to Image and Vision Computing, Special Edition; The use of Probabilistic Models in Computer Vision.

5. P.A.BROMILEY, N.A.THACKER AND P.COURTNEY, *Non-Parametric Subtraction Using Grey Level Scattergrams*, BMVC 2000, Bristol, pp 795-804, Sept. 2000.

6. G. COWAN *Statistical Data Analysis,* Oxford University Press, ISBN 0-19-850156-0, 1998.

7. A.P. DAWID, *Probability Forecasting*, Encyclopedia of Statistical Science 7, pp 210-218. Wiley, 1986.

8. A. LORUSSO, D.W. EGGERT, AND R.B. FISHER, *Estimating 3D Rigid Body Transformations: A Comparison of Four Algorithms*, Machine Vision Applications, 9 (5/6), 1997, pp.272-290.

9. W. FOERSTNER, *10 Pros and Cons Against Performance Characterisation of Vision Algorithms*, Proceedings of ECCV Workshop on Performance Characteristics of Vision Algorithms, Cambridge, UK, April 1996. Also in Machine Vision Applications, 9 (5/6), 1997, pp.215-218.

10. R.M. HARALICK, *Performance Characterization in Computer Vision*, CVGIP-IE, 60, 1994, pp.245-249.

11. R.M. HARALICK, C.N. LEE, K. OTTENBERG AND M. NOELLE, *Review and Analysis of Solutions to the Three Point Perspective Pose Estimation Problem*, Intl. J. Computer Vision, 13(3), 1994, pp.331-356.

12. M.D.RICHARD AND R.P.LIPPMANN, *Neural Network Classifiers Estimate Bayesian a Posterioi Probabilities*, Neural Computation,3,461-483,1991.

13. S.J. MAYBANK, *Probabilistic Analysis of the Application of the Cross Ratio to Model Based Vision*, Intl. J. Computer Vision, 16, 1995, pp.5-33.

14. D. MARR, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information* Publisher W. H. Freeman Company, NY 1982.

15. P. MEER, D. MINTZ, A. ROSENFIELD AND DONG YOOM KIM *Robust Regression Methods for Coputer Vision: A Review* Intl. J. Computer Vision, 6:1, 1991, pp. 59-70.

16. J. NEYMAN, *X-Outline of a Theory of Statistical Estimation Based on the Classical Theory of Probability*, Phil. Trans. Royal Soc. London, **A236**, pp. 333-380, 1937.

17. P. J. PHILLIPS, A. MARTIN, C. L. WILSON AND M. PRZYBOCKI, *An Introduction to Evaluating Biometric Systems* IEEE Computer Special Issue on Biometrics, pp. 56-63, Feb. 2000.

18. I. POOLE, *Optimal Probabilistic Relaxation Labeling*, Proc. BMVC 1990, BMVA, 1990.

19. W. H. PRESS, B. P., FLANNERY, S. A. TEUKOLSKY AND W. T. VETTERLING, *Numerical Recipies in C* Cambridge University Press., 1991

20. NUMERICAL RECIPIES IN C

21. G. REES, P. GREENWAY AND D. MORRAY, *Metrics for Image Segmentation*, Proceedings of ICVS Workshop on Performance Characterisation and Benchmarking of Vision Systems, Gran Canaria, January 1999.

22. A. Roche, G.Malandain, N.Ayache and S.Prima. Towards a Better Comprehension of Similarity Measures Used in Medical Image Registration. MICCA, 555-566, 1999.

23. R.S. STEPHENS, *A Probabilistic Approach to the Hough Transform*, British Machine Vision Conference BMVC90, 1990.

24. A. STUART, K. ORD AND S. ARNOLD *Kendall's Advanced Theory of Statistics* Vol. 2A, Classical Inference and the Linear Model, Sixth Edition, Arnold Publishers, 1999.

25. HTTP://WWW.TINA-VISION.NET/ *TINA: Open Source Image Analysis Environment* ISBE, University of Manchester, UK

26. N A THACKER AND J E W MAYHEW, *Optimal Combination of Stereo Camera Calibration from Arbitrary Stereo Images* Image and Vision Computing, Vol. 9 No. 1, pp. 27-32, 1991.

27. N. A. THACKER, F. J. AHERNE AND P. I. ROCKETT, *The Bhattacharyya Metric as an Absolute Similarity Measure for Frequency Coded Data*, Kybernetika, Vol. 32 No. 4, pp. 1-7, 1997

28. N.A.THACKER, F.AHEARNE AND P.I.ROCKETT, 'The Bhattacharryya Metric as an Absolute Similarity Measure for Frequency Coded Data.' Kybernetika, 34, 4, 363-368, 1997.

29. P. COURTNEY, N.A.THACKER AND A.CLARK, *Algorithmic Modelling for Performance Evaluation*, Machine Vision and Applications, 9, 219-288, 1997.

30. N.A.THACKER AND A.J.READER, *Modal Division and its Application to Medical Image Analysis*, Proc. MIUA, pp 7-10, London. 10th-11th July, 2000.

31. Paul Viola, Alignment by Maximisation of Mutual Information, M.I.T. PhD Thesis, 1995.

32. N. WIENER, *Extrapolation, Interpolation and Smoothing of Stationary Time Series with an Appendix by N. Levinson* Technology Press of the MIT and J. Wiley, New York, 1949.