Tina Memo No. 1996-001
Short version presented at: ECCV, Cambridge, 1996.

Algorithmic Modelling for Performance Evaluation.

N.A.Thacker, P.Courtney and A.Clark.

Last updated
1 / 12 / 1998

This document forms part of the **Features and Measurement** Series available from www.tina-vision.net.

2002-005	Tutorial: An Empirical Design Methodology for the Construction of Machine Vision Systems.
1996-001	Algorithmic Modelling for Performance Evaluation.
2006-002	A Statistical Framework for Detection of Connected Features.
2006-007	Quantitative Verification of Projected Views Using a Power Law Model of Feature Detection.
1997-003	Tutorial: Supervised Neural Networks in Machine Vision.
1995-003	Invariance Network Architecture.
1992-001	Combining the Opinions of Several Early Vision Modules using a Multi-Layer Perceptron.
2005-006	Curve Fitting and Image Potentials: A Unification within the Likelihood Framework.
1996-002	Tutorial: The Likelihood Interpretation of the Kalman Filter.
1994-003	Using a Switchable Model Kalman Filter.
2004-012	Tutorial: Computing 2D and 3D Optical Flow.
2005-011	Comparing the Performance of Least-Squares Estimators: when is GTLS Better than LS?
1994-001	Tutorial: Overview of Stereo Matching Research.
1992-002	Online Stereo Camera Calibration.
1995-002	Calibrating a 4 DOF Stereo Head.
2000-009	An Evaluation of the Performance of RANSAC Algorithms for Stereo Camera Calibration.
2001-011	The Evolution of the TINA Stereo Vision Sub-System.
2007-011	A Methodology for Constructing View-Dependent Wireframe Models.

WWW.TINA-VISION.NET

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

1 Introduction

Many of the vision algorithms described in the literature are tested on a very small number of images. It is generally agreed that algorithms need to be tested on much larger numbers if any statistically meaningful measure of performance is to be obtained. However, these tests are rarely performed; in our opinion this is normally due to two reasons. Firstly, the scale of the testing problem when high levels of reliability are sought, since it is the proportion of failure cases that allows the reliability to be assessed and a large number of failure cases are needed to form an accurate estimation of reliability. For reliable and robust algorithms, this requires an inordinate number of test cases. Secondly, the difficulty of selecting test images to ensure that they are representative. This is aggravated by fact that assumptions made may be valid in one application domain but not in another. This makes it very difficult to relate the results of one evaluation to other users' requirements.

What we suggest is a methodology for algorithmic testing whereby, instead of attempting to model the data itself, the statistical data distributions which effect algorithm performance are identified and evaluation performed by modelling the algortithm. Once a method of obtaining and evaluating the performance of the algorithm based on these distributions has been developed, the algorithm can be rapidly re-evaluated for any new image data set specification. It is the evaluation method itself, rather than a set of performance measures specific to one data set, which then provides the measure of algorithmic performance. Often the combinatorial advantages obtained by working directly with data distributions also results in a requirement for far fewer test images. This technique is demonstrated on algorithms for feature detection, stereo matching and view-based object recognition which operate on image and other kinds of input data.

We suggest that in order to adopt this methodology within the research arena, we will first need to agree measures for performance quantification for common groups of algorithms. The process of then applying these evaluation procedures to algorithms could then be aided greatly by the use of standard test dataset groups and appropriate software harnesses.

2 The Need for Algorithmic Evaluation

A meaningful methodology for algorithmic evaluation is needed for at least two reasons: to demonstrate the capabilities of an algorithm in a particular application and thus estimate its effectiveness; and to provide a systematic method for evaluating (perhaps incremental) changes to algorithms. There has been much good work in the past few decades in the development of algorithms for the extraction of various types of information from images. This work has generally concentrated on the assumptions that must be made regarding the data and the numerical form of possible solutions. Often quite strong assumptions as to the characteristics of the data are imposed for reasons of mathematical tractability. Much less work has been published on the systematic evaluation of algorithms in terms of relaxing these assumptions or for the purposes described above. This may in part be due to the fact that a rigorous evaluation is a large amount of extra work and has often not been perceived as publishable in the same way as a novel piece of mathematics or the demonstration of a new application.

Systematic evaluation may require a completely different approach to algorithm design and testing. Having developed an algorithm over a period of several years, the complexity of the resulting algorithm may be such that it is impossible to evaluate the algorithm in any way other than treating it as a black box. As complexity increases, it becomes progressively harder for such a black box evaluation to provide accurate performance predictions to a potential user. This is caused by the increased number of possible discontinuities due to special cases likely to be present. Furthermore, if the user intends to use the algorithm as a part of a larger automatic system, the quality of the output data needs to be suitable for the next stage in the system. Indeed, it may be argued that if the system is to be used in an application with any social or economic value, a simple algorithm with predictable performance may be better than a complex algorithm with better mean but less predictable performance. In short, algorithms need to deliver not only the answer but accuracy and confidence estimates if the data are to be used reliably in a system. Any algorithm that can only be evaluated as a black box may never produce such output. This suggests that the way to develop good algorithms is to perform algorithm evaluation hand-in-hand with increasing complexity, adding new stages to an algorithm only when the effects of the change can be adequately modeled. This is a more rigorous, though perhaps slower, approach to algorithmic development than is generally followed.

3 Alternative Approaches to the Problem

Computer vision is the process of extracting useful information from images in order to perform a specific task. This practical emphasis is often forgotten in some algorithmic research but is an important part of the definition. Clearly, algorithms which deliver information that is of no practical use will never be used. The first requirement of algorithmic research is therefore to specify the information that we wish to obtain from the image. We can generally expect that once this information has been obtained it will be used as the basis for subsequent action resulting from a decision based on this data.

Given that we wish to determine information for a particular purpose, we now need to know whether there is an optimal way of presenting it. Clearly, a decision-making process based on delivered data will crucially require information regarding the expected success of a particular outcome given the data. There are two ways that the successful outcome can be affected: the first by a failure in action (the answer was correct but the decision was wrong) and the second an error in the data (the answer was wrong so a wrong decision was made). As a consequence, an algorithm must not only deliver an estimate of the required data but also an estimate of data reliability. Anything less than this will make subsequent decision formation unreliable and the algorithm can only form part of a practical system under very restricted circumstance. The most common form in which confidence is represented is as an error covariance measure.

The most direct information regarding the successful outcome of a particular decision is the posterior (Bayes) probability. This is defined as the probability that a particular event will be true given a particular observation. Knowledge of Bayes probabilities of outcomes given a set of alternative states allows a direct assessment of attempting alternative actions. Combined with the cost and benefits of making various decisions under the various states allows system performance to be ascertained and optimised. Probability theory is regarded as the only self-consistent computational framework for all data analysis and decision making. It is therefore not surprising that it forms the basis of all statistical analysis processes.

There have been a number of past attempts to address the thorny subject of performance evaluation of algorithms. These are discussed below.

The analytic approach. The most obvious approach is to use propagate the distributions through the various stages of an algorithm analytically. This was, for example, undertaken successfully by Maybank for the cross-ratio [1]. However, the approach quickly becomes untractable, even for simple algorithms.

Propagation of covariances. This approach was proposed by Haralick [2] and Foerstner [4]. It is again tractable for relatively simple algorithms; however, as currently developed, the theory is restricted to linearisable algorithms and simple distributions which can be expressed in a few parameters.

Repeated testing. Repeated testing of an algorithm can be used to build a statistical description of its performance—see [5] for a recent example of this approach. The biggest problem with repeated testing is the excessive amounts of testing involved, making impractical for use with all but the fastest algorithms.

We believe that a fourth alternative may provide a complementary solution and follows on directly from our statistical interpretation of algorithms. This approach is outlined in the following section.

4 The Algorithmic Modeling Approach

There seem to be at least two ways that statistics can play a role in computer vision. The first is in the construction of algorithms on the basis of statistical assumptions regarding the data, while the second is in the testing of the validity of these assumptions and quantification of algorithmic performance. An evaluation procedure which attempts to make both of these stages explicit would provide a complete specification of the performance of an algorithm. The methodology for algorithm evaluation which we would advocate in the general case would do exactly this: firstly, identify the data distributions that are key to the statistical assumptions embodied in the algorithm; secondly, sample these distributions for a test set of data; and finally, build a model of the algorithm which allows the shapes of these distributions to be propagated through to describe effects on the output performance.

This approach differs significantly from that proposed by Haralick in [3] which places the emphasis on data modeling. We propose that the data distributions should be sampled and that the place for modeling is in the description of the algorithm itself. The data from the world are, after all, an unknown quantity with such diversity that any predictions about their behaviour can never be guaranteed in advance for an algorithm with heuristic components and may change with a respecification of the target data set. The algorithm on the other hand is known, which makes it possible to construct a computational model of it. There is then a separate stage of matching data sources to modeled algorithms.

Access to a simple computational model of an algorithm leads to new possibilities for algorithm evaluation and development. A simple prescription of the methods required to extract the statistical distributions from arbitrary images and a computational model of the algorithm are all that would be required for any potential user of the algorithm to perform his own evaluation for his own data set. Algorithm performance may even be predicted for an individual image, thus coming closer to the system requirements for algorithms. Algorithmic models allow the analysis of the data-independent properties of the algorithm. Algorithmic models also get away from the difficult problem of not being able to produce meaningful comparisons of alternative algorithms. If the algorithm model has been designed by its original authors, then it must encapsulate a fair estimate of the expected performance of the algorithm as understood by those that designed it. It also provides an independent estimate of expected performance for people wishing to develop their own implementation of the algorithm which also allows an additional level for scientific verification of the original work.

5 Practical experience of algorithm modeling

Over the past few years we have been applying this approach to examine the performance of several algorithms for vision tasks in feature detection, matching, object recognition and others [15].

6 Modeling of a feature detection algorithm

In a recent industrial application of vision, it was necessary to automatically capture the trajectory of a high speed moving object. Several image processing techniques were considered, including corners [6]. These are attractive for 2D image plane tracking as since they do not suffer from the aperture problem in the way that line features do. There are numerous reference to the successful application of corners to vision tasks [8] [9] [10]. However initial trials at ITMI showed that corners were poorly detected from sample images. It was clear that the reliability of corner extraction depended in some way on image characteristics but this precise relationship had not been thoroughly investigated. A study was therefore carried out to address this issue and to determine under what conditions can corners be reliably extracted, whether this reliability can be quantified and what can be done to improve this reliability (for further details and experimental validation see [11]).

A corner detector typically takes a region of an image around a point and returns a boolean corner/non-corner response. This gives rise to two kinds of classification errors: not signaling a corner when there is one (non-detection), and signaling a corner when there is none (false detection). Depending upon the application cost of such errors (which are in general different), there is an optimum trade-off between non-detection and false detection.

Internally such a "black box" detector computes a function which is an operation on the grey level values of a group of pixels. The resulting value is then compared to a threshold to make the classification detection. Although there are often many parameters within the detector, it is the choice of this threshold which is critical to optimising the trade-off. Without knowing the application-level costs of the two kinds of errors, it is therefore essential to know the probability distributions of the corner strengths in the two cases: when there is a corner (the corner pdf); and when there is no corner (the non-corner pdf).

6.1 constructing the corner and non-corner pdfs

A simulation was carried out to construct these two pdfs. Three 3x3 pixel binary datasets were created to represent the cases of flat background, straight edge and corner of 90 degrees. These datasets were then fed into the corner detector to yield the nominal corner strength estimates of background (0.0), edge (-7.2) and corner (48.8).

A series of 10,000 noisy background datasets were generated by adding uniform noise of magnitude $+e \ldots - e$ independently to each pixel in the original background datasets. These were then fed into the detection algorithm under test and the elementary background response Pb pdf built up for that value of e. This process was repeated for the edge and corner datasets to construct the edge Pe and corner Pc pdfs. A table of such elementary pdfs were constructed for varying values of e from 1/256 (typically the best attainable with an 8 bit grey level representation) to 1/2 (severe noise SNR of 0dB). These pdfs tend to appear symmetric bell-shaped for weak amounts of noise, with a strong asymmetry becomes evident with increasing noise magnitudes, especially when e exceeds 1/8.

Although with the corner strength estimates were computed for the pixel at the center of each dataset, those pixels adjacent to the center pixel also give rise to corner strength estimates. These give rise to their own elementary pdfs which can also be estimated in the same way.

6.2 using the model to estimate performance

Having obtained these elementary pdfs for all cases on background, edge, corner, as well as near-edge and near-corner pixels, it is possible to combine them to create a composite non-corner pdf. A false detection occurs when a corner strength drawn from this non-corner distribution exceeds the preset corner threshold. Likewise corner non-detection occurs when the corner strength drawn from the corner distribution falls below the preset corner threshold. (Note that including all near-corner pdfs in this composite implicitly imposes a corner localisation precision of one pixel.)

Image regions in real images differ from the simulation templates which effects the elementary pdfs in the following ways: **non-unit contrast:** since the corner detection is a double difference, the corner strength is the fourth power of the image contrast. The elementary pdf can be scaled in x before being added to the composite to take this into account. **non-uniform noise:** the corner detector operates in on several pixel values and the output distribution should approximate a gaussian, in accordance with the central limit theorem. This will apply equally well to alternative noise models. The shape of the observed small-noise pdfs for uniform noise is in occordance with the CLT. **correlation between pixels:** further trials with smoothed noisy regions showed that the corner strength is strongly attenuated, especially when the correlation exceeds 0.2. Pixel resampling is required to restore corner strength and thus detectability, but this results in a reduction in localisation. This is the detection-localisation trade-off identified by Canny [7].

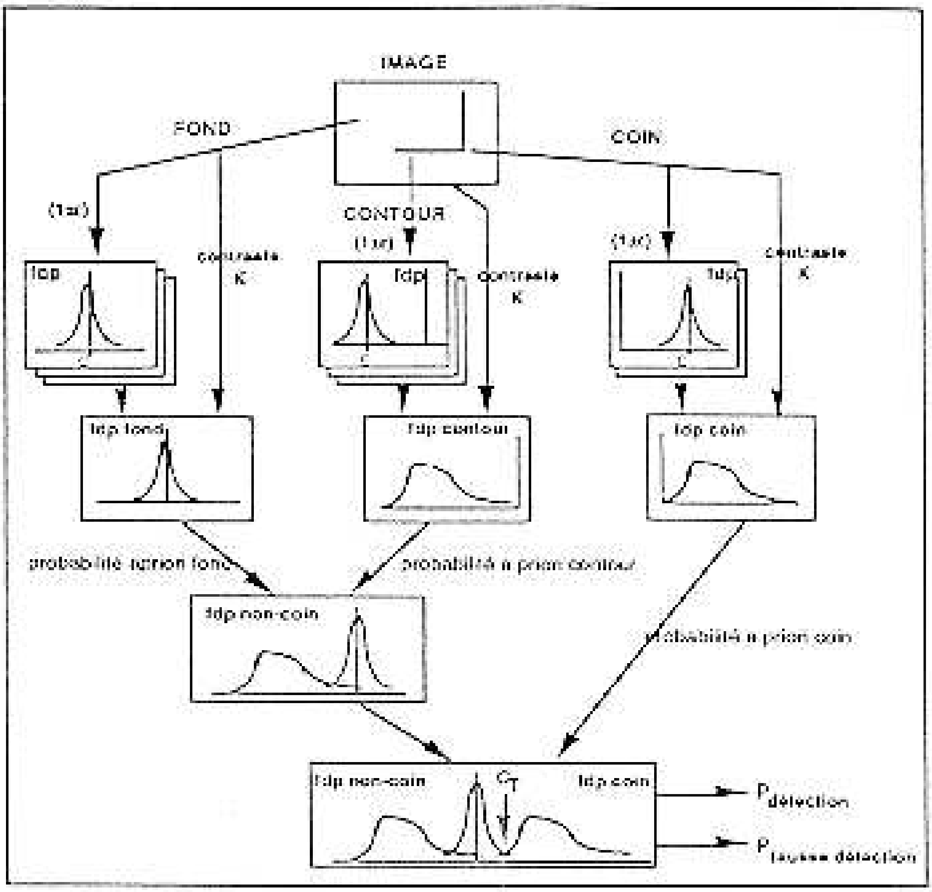

Figure 1: Overview of the Proposed Method

Furthermore, the prior probabilities of the elementary events background, edge, near-edge and near-corner can be taken into account when creating the composite non-corner pdf. Although they may not be known precisely, rough values are sufficient to give a give the relative probabilities of non-detection and false detection for a given threshold.

Figure 1 shows how the elementary pdfs may be combined with image parameters of contrast, noise magnitude and prior distribution of corners, edges and background to compute the non-detection and false detection pdfs. This allows us to establish the conditions under which corners can be reliably extracted in terms of signal-to-noise ratio, interpixel correlations etc. It also allows us to determine not only the optimum threshold, but also the two failure rates and thus the application-level costs.

7 Modeling of a stereo matching algorithm

One common vision subtask is the matching of image features for scene reconstruction from stereo and motion estimation. Such matching procedures generally make use of several heuristics such as local image similarity (using for example a cut ρ on the cross correlation, and ω on the ratio of absolute feature strengths), restricted search areas A, smoothness, one-to-one matching (using for example a uniqueness cut δ) and so on. Furthermore these heuristics interact with the characteristics of the feature detector, and in particular its reliability under noise or occlusion in the scene. This strongly determines the performance of the matching algorithm and these effects were studied for a corner based matcher in [12].

7.1 constructing the mismatch and signal rejection distributions

We can analyse the conditions under which we will get a mismatch and reject a correct match by considering each corner feature and its available match candidates in turn. We will thus show how the probabilities of accepting noise and rejecting signal can be controlled by the parameters of the matching algorithm (ρ, ω, A and δ). Note that in the following analysis we assume that the cross correlation distributions for correct and incorrect matches are independent of the detection process. For a given image region there are several cases which can arise:

incorrect match case (a): This is the case that the cross-correlation for the best candidate match x_m to the corner under consideration ("current") is incorrect in the case where neither candidate has their pair detected in the other image.

The probability P_m^a for the best candidate match to the current corner being one of n_u unpaired random corners can be computed as a product of several factors so:

$$P_m^a = 2n_u P_d^2 (1 - P_d)^2 P_I(\rho)$$

where P_d is the probability of detection $P_{I(\rho)}$ is the probability that a random match drawn from the cross correlation for incorrect matches P_N will have a value greater than ρ:

$$P_I(x) = \int_x^1 P_N(a)da$$

incorrect match case (b): The case that of obtaining an incorrect match with one of n_u unpaired random corners when the correct match to one of the corners was also present. This is slightly more complicated than the previous case because the existence of the correct match in the matching list may still prevent this getting accepted as a match due to the uniqueness parameter δ. The probability of this happening P_m^b is given by:

$$P_m^b = 4n_u P_d^3 (1 - P_d) P_n(\delta, \rho)$$

where $P_n(\delta, \rho)$ is the probability that an incorrect match can be chosen even when the correct match is present in the match list above a value of ρ. Given that P_N and P_S (the cross correlation distribution for correct matches) are uncorrelated this is given by:

$$P_n(\delta, \rho) = \int_\rho^{1-\delta} P_S(x) \int_x^{1-\delta} P_N(a - \delta) da dx$$

incorrect match case (c): Where the current corner is paired but has been matched incorrectly with one of $2n_p$ paired random corners. We may wish to write the probability for the acceptance rate for mismatches P_m^c as:

$$P_m^c = 2n_p P_d^4 P_n(\delta, \rho)^2$$

This equation assumes that the two probabilities for mismatch $P_n(\delta, \rho)$ are uncorrelated which is unrealistic, as the two corners must be figurally similar if they are to have mismatched in one matching direction. Thus it is better to write this as:

$$P_m^c = 2n_p P_d^4 P_n(\delta, \rho) P_k(\delta, \rho)$$

where $P_k(\delta, \rho)$ is the probability that the cross correlation value for the complementary pair of the original random match will also be bigger than the correlation value for the correct match.

We now consider ways in which corner pairs are rejected by the matching process.

rejected match case (a): The first case we consider is when the current corner has been detected in both images (ie paired) and a random matching feature has not been detected in either image. The probability of rejecting this match is given by:

$$P_r^a = P_d^2(1 - P_d)^2 P_J(\rho)$$

where

$$P_J(\rho) = \int_0^\rho P_S(x)dx$$

rejected match case (b): When the current match is paired and there is a random un-paired match present in either image. The probability of rejecting a correct match due to the presence of n_u unpaired random corners P_r is given by:

$$P_r^b = 2P_d^2(1 - P_d)^2(P_J(\rho) + n_u P_l(\delta, \rho))$$

where $P_l(\delta, \rho)$ is the probability of rejecting a correct match due to the proximity of a random corner.

$$P_l(\delta, \rho) = \int_\rho^1 P_S(x) \int_x^{1+\delta} P_N(a + \delta)dadx$$

rejected match case (c): The final case for consideration is when the current match is paired and there is a random paired match. The rejection rate for good corner matches P_r^c is given by

$$P_r^c = P_d^4(P_J(\rho) + 2n_p P_l(\delta, \rho) - n_p^2 P_l(\delta, \rho)^2)$$

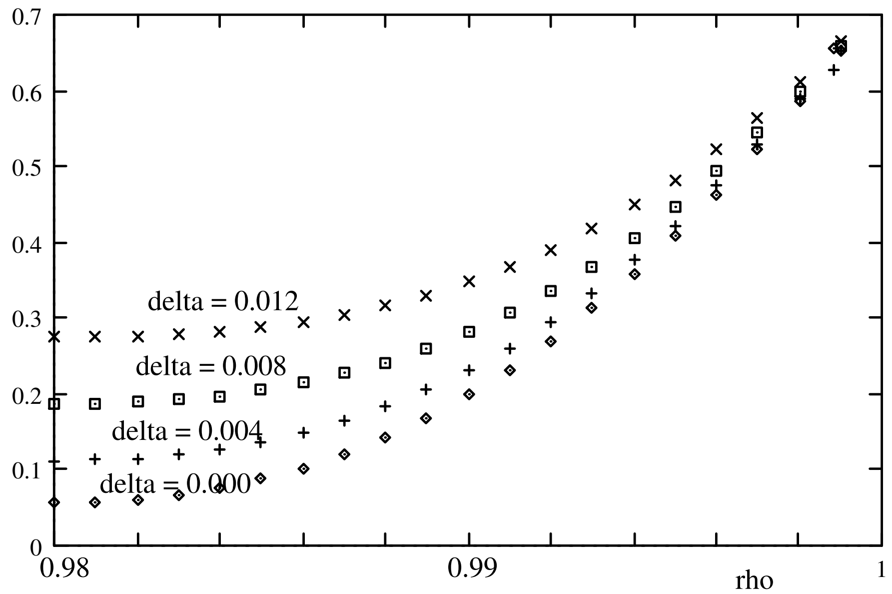

Figure 2: Quantity of Rejected Data

The cross-correlation distributions $P_N(x)$, $P_S(x)$ and $P_k(\delta, \rho)$ can be approximated by triangular distributions. The remaining unknown parameters are the detection efficiency P_d and the numbers of paired and unpaired random corners n_p and n_u. In some ways these values are closely related as the ratio $n_p : n_u$ has a maximum value of $P_d : 1 - P_d$. In an application where the full image contains several hundred corners and the search regions are of the order of a few percent of the image we estimate these values as $P_d = 0.85, n_u = 0.75, n_p = 4.25$ and we can use these values we can now compute typical signal rejection and noise acceptance curves for the matching algorithm as a function of the matching parameters ρ and δ. See figures 2 and 3.

7.2 using the model to estimate performance

We can thus approximate the total number of incorrect matches as:

$$P_m^T = P_m^a + P_m^b + P_m^c$$

and the total rejection rate for paired corners as:

$$P_r^T = P_r^a + P_r^b + P_r^c$$

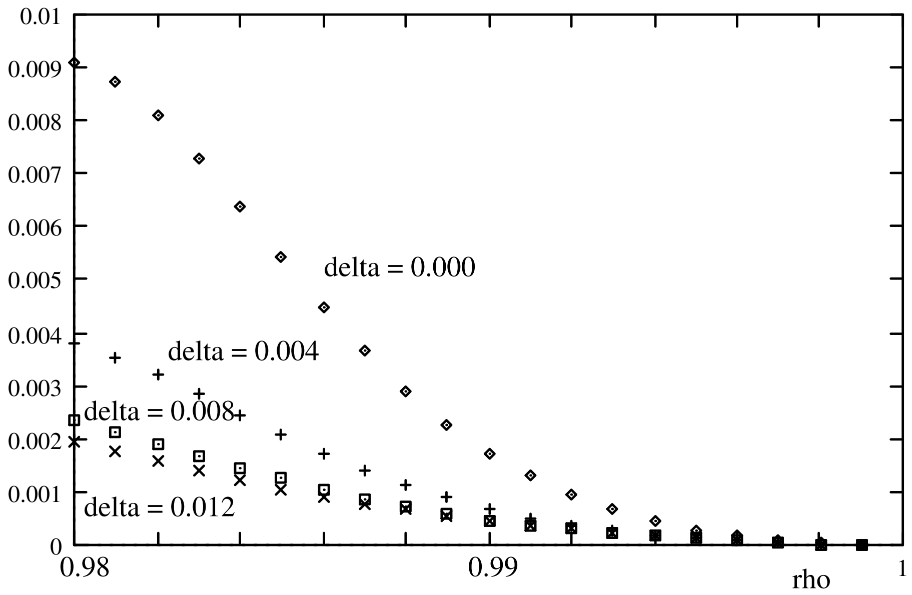

Figure 3: Quantity of Miss-matched Data

In a specific application where the detected corners have correlated properties, the probability distributions for cross correlations of signal and background may be significantly different. In these cases the probabilities for mismatch and signal rejection would also be different. However, we can still draw some qualitative conclusions about the generic case of corner matching which must be true regardless of the signal and background distributions. These are:

(1) All terms in P_m^T are proportional to the mean number of candidate matches, thus we would expect the total number of mismatches to vary proportionately with the search area A.

(2) We expect type (a) mismatches to be a very small fraction of the total number of mismatches. The only way to remove these is to increase the minimum required cross correlation value ρ.

(3) We expect type (b) and (c) mismatches to be of roughly equal importance and both are reduced considerably by use of the uniqueness parameter δ at the cost of only marginal reduction in the overall number of matches.

(4) There is no improvement obtained by increasing δ beyond a value of $1 - \rho$ as at this point all mismatches of type (b) have already been rejected.

(5) There is no set of parameters which give an optimal signal to noise ratio, this value keeps on rising with increasing ρ. There are however optimal values of ρ and δ corresponding to the minimum noise obtainable for a required proportion of signal. For example using the above model for the data the minimum noise obtainable at a signal level of 60% is 0.2% at parameter values of $\rho = 0.985$ and $\delta = 0.0032$.

(6) Even in very severe cases we expect this matching algorithm to have a signal to noise ratio in excess of 100:1.

8 Modeling of an object recognition algorithm

Another example of algorithm modeling has been used in the evaluation of object recognition using Pairwise Geometric Histograms (PGH). A PGH encodes local shape geometry in a manner which is invariant to rotation and translation and is robust to missing data line, fragmentation and scene clutter [13]. We have recently modified the algorithm to cope with scaled objects [16]. The representation is also complete, in that it is possible to reconstruct a shape from the set of histograms which describe it [17]. The statistical distribution which defines the performance of this technique is the "Match score frequency distribution". This is the frequency distribution of cross match scores (using the Bhattacharyya metric) between all histograms representing edge segments from objects in the model data base. The probability that a random incorrect match score will lie within a given range can be obtained by integration of this distribution (figure 4). This gives a direct estimate of the expected mismatch rates for various alternative forms of the algorithm, including for example histogram size and bin numbers (figure 5). The major reductions in the the match score from the maximum value of 1.0 will be due to effects such as object occlusion and scene clutter. These can be estimated as

$$\delta D \;=\; Dk$$

where k is the proportion of missing data, and

$$\delta D \;=\; Daf/(1+af)$$

where a is the quantity of scene clutter and f is the mean proportion of non zero entries in the PGH. Thus we are able to directly estimate the effects of clutter and occlusion on our algorithm. In addition knowledge of this distribution allows us to go on to use standard statistical pattern recognition techniques to show that [14]:

- The reliability of recognition for the entire algorithm is unaffected by model data base size.
- The method can represent very large numbers of distinct shapes.
- The processing requirement of the algorithm scales in a manner which is no worse than linear with the number of stored models.

Taking this proces one step further and estimating the probablilty of a mismatch as a function of the fraction of reconstructed match signal would give us a performance evaluation which is completely independent of the match metric used and allows a cross comparison of different statistical matching techniques, such an evaluation has also been used in [15].

9 Discussion and Conclusions

There are several ways of computing posterior probabilities and directly related quantities under limited circumstances. However, in many practical situations problems cannot easily be formulated to correspond exactly to a particular computation. Compromises have to be made, generally in assumptions about the statistical form of the processed data, and it is the adequacy of these compromises that will determine the success or failure of a particular algorithm. Clearly therefore, understanding these assumptions and compromises is an important part of algorithmic development. We can conclude that algorithms that will work best on a particular application are those that model most closely the underlying statistics of the measurement process and correctly propagate the effects of these through to the output of the algorithm. Algorithmic robustness goes hand in hand with getting this process correct. It can only be compromised in the interest of computational speed if the trade-off is understood and the consequences accepted. Sadly, this trade-off is usually made in ignorance of the issues involved. One of the major criticisms of computer vision over the past few years has been due to a general lack algorithmic reliability. This has largely been due to the neglect of the important role that statistics must play in algorithm development. Indeed, one could go as far as to say that computer vision should strictly be regarded as a branch of applied statistics.

Access to a simple computational model of an algorithm leads to new possibilities for algorithm evaluation and development. A simple prescription of the methods required to extract the statistical distributions from arbitrary images and a computational model of the algorithm are all that would be required for any potential user of the algorithm to perform his own evaluation for his own data set. Algorithm performance may even be predicted for an individual image, thus coming closer to the system requirement of algorithms with predictable performance. Algorithmic models allow the analysis of the data-independent properties of the algorithm. Algorithmic models get away from the difficult problem of not being able to produce meaningful comparisons of alternative algorithms. If the algorithm model has been designed by its original authors, then it must encapsulate a fair estimate of the expected performance of the algorithm as understood by those who designed it. It also provides an independent estimate of expected performance for people wishing to develop their own implementation of the algorithm which also allows an additional level for scientific verification of the original work. The commonality of this evaluation strategy across a wide variety of algorithms has prompted us to suggest the use of a common software harness as a framework for algorithm testing.

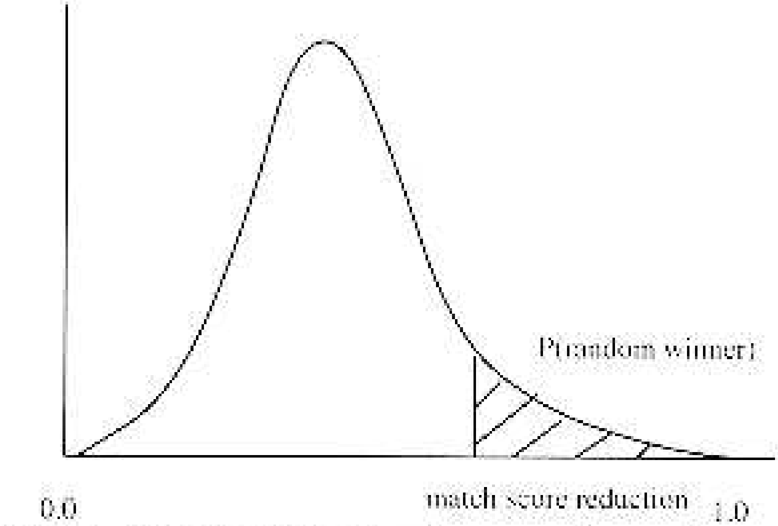

Figure 4. Probability of geting a random winner.

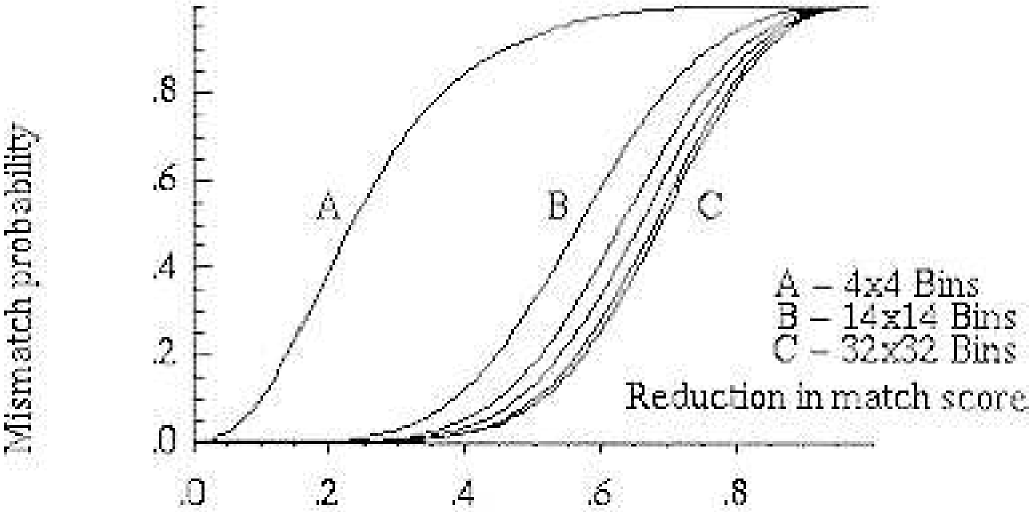

Figure 4: Reduction in Match Score

References

[1] Maybank, S. 'Error trade-offs for the Cross-Ratio in Model Based Vision', Workshop on Computer Vision for Space Applications, Antibes pp. 350-359, 1993.

[2] Haralick, R.M., 'Propagating Covariances in Computer Vision', ICPR94, Jerusalem, Sept. 1994.

[3] Haralick, R.M., 'Performance Characterisation Protocol in Computer Vision', Image Understanding Workshop, Monterey, California, 1995.

[4] Foerstner, W., 'Diagnostics and Performance evaluation in Computer Vision', Workshop on Robust Computer Vision, Seattle 1994.

[5] Nielsen, J., ECVnet Workshop on Benchmarking, Paris 1995.

[6] Harris, C. and M.Stephens, 'A Combined Corner and Edge Detector', Proc. 4th Alvey Vision Conference, pp.147-151, 1988.

[7] Canny, J.F., 'A Computation Approach to Edge Detection', IEEE PAMI, 8, pp. 679-698, 1986.

[8] Thacker, N.A., Y.Zheng and R. Blackbourn, 'Using a Combined Stereo/Temporal Matcher to Determine Ego-motion', Proc. BMVC, pp.121-126, 1990.

[9] Thacker, N.A. and J.E.W.Mayhew, 'Optimal Combination of Stereo Camera Calibration from Arbitrary Stereo Images', Image and Vision Computing, pp.27-32, vol 9 no 1, Feb. 1991.

[10] Wang, H. and J.M.Brady, 'A Structure-from-Motion Algorithm for Robot Vehicle Guidance', Proc. IEE Symp. Intel. Vehic., Detroit, June 1992.

[11] Courtney, P. and Skordas, Th, 'caracterisation de performances des algorithmes de vision - un exemple: le detecteur de coins', AFCET-RFIA 1996 (to appear).

[12] Thacker, N.A. and Courtney, P., 'Statistical Analysis of a Stereo Matching Algorithm', Proc. BMVC, pp.316-326, 1992.

[13] A.P.Ashbrook, N.A.Thacker and P.I.Rockett, 'Pairwise Geometric Histograms: A scalable solution for the Recognition of 2D rigid shape', Proc. SCIA95, June 1995.

[14] A.P.Ashbrook, P.I.Rockett and N.A.Thacker, 'Multiple Shape Recognition using Pairwise Geometric Histogram Based Algorithms', IEE Image Processing, Edinburgh, July 1995.

[15] R.Lane, N.A.Thacker and N.L.Seed,'Stretch Correlation as a Real-Time Alternative to Feature Based Stereo Matching Algorithms', Image and Vision Computing, 12,4, pp 203-212, 1994.

[16] A.P.Ashbrook, N.A.Thacker, P.I.Rockett and C.I.Brown, 'Robust Recognition of Scaled Shapes Using Pairwise Geometric Histograms', BMVC95, Birmingham, 1995.

[17] N.A.Thacker, P.A.Riocreux, and R.B.Yates, 'Assessing the Completeness Properties of Pairwise Geometric Histograms', Image and Vision Computing, 13, 5, 423-429, 1995.

[18] Ousterhout, J.K., Tcl and the Tk Toolkit, Addison-Wesley, 1994.

[19] Atick, J.J., Griffin, P.A. and Redlich, A.N., 'Face Recognition from Live Video for Real-World Applications', Advanced Imaging, 10(5), pp. 58-62, May 1995.

[20] Libes, D., 'Exploring Expect: A Tcl-based Toolkit for Automating Interactive Programs', O'Reilly and Associates, 1994.