Performance Characterisation in Computer Vision: The Role of Statistics in Testing and Design

P. Courtney* N.A. Thacker†

August 29, 2003

Abstract

We consider the relationship between the performance characteristics of vision algorithms and algorithm design. In the first part we discuss the issues involved in testing. A description of good practice is given covering test objectives, test data, test metrics and the test protocol. In the second part we discuss aspects of good algorithmic design including understanding of the statistical properties of data and common algorithmic operations, and suggest how some common problems may be overcome.

1 Introduction

Some 30 years of research has produced a rich variety of methods for processing image data, but little information on how they perform beyond a few example images. Presenting results as images rarely conveys any statistically useful measure of performance, partly because the choice of test images is not justified with any rigour, and partly because such results are in a qualitative and subjective form. It is therefore still very hard to reuse those algorithms that are described in the literature, let alone to build systems.

One of the criteria that appears to dominate the development of new algorithms for publication is novelty and sophistication. However, as the sophistication of an algorithm increases it seems likely that the specificity of the assumptions embedded within it will increase. This gives rise to Bowyer's conjecture [1] that performance will tail off or even fall as sophistication increases and suggests that consideration of the underlying assumptions and testing must be an essential part of algorithm development.

Many objections have been raised against carrying out empirical work in machine vision (see [2] for an extensive list with counter arguments). One objection in particular seems especially prevalent: that performance depends on the application and so cannot be studied independently of it. Whilst the actual *choice* of which algorithm to use can generally only be made in the context of a particular task, the *performance characteristics* of the algorithm may be described. If we consider another discipline, such as electrical engineering, we realise that a component such as a transistor can be used in an almost infinite range of circuits. However, transistors are specified in terms of a limited range of agreed measures such as linearity, stability, sensitivity and power handling ability, as well as speed, size, packaging and cost.

*Visual Automation Ltd, Manchester, England (`patrick.courtney@acm.org`).

†Division of Imaging Science and Biomedical Engineering, University of Manchester, England.

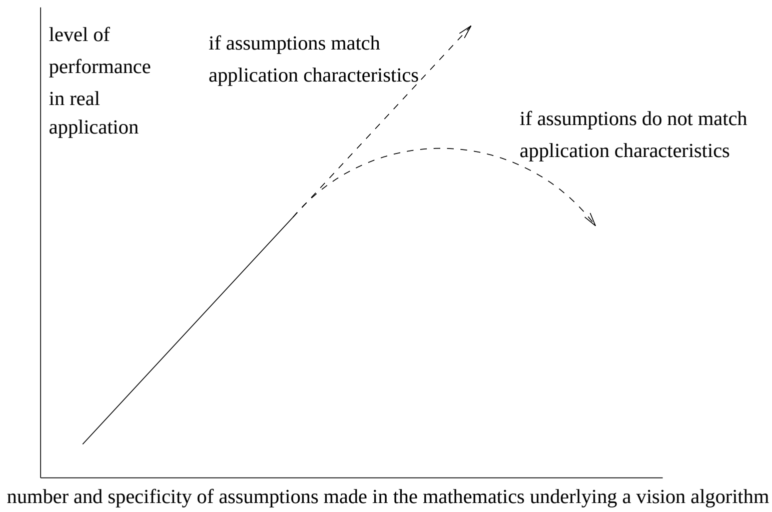


Figure 1: Performance as a function of mathematical sophistication [1]

This allows them to be used in a range of applications (switching, amplification, detection etc). This is the situation for any engineering discipline. Similarly, vision algorithms can be described in terms of abstract conditions such a contrast, noise, object geometry etc. The generalisability of such measures then becomes an important question for system building and design.

Whilst there has been some debate about the relationship between theory and empirical work, empirical work is clearly part of the scientific method and is required in order to provide a solid experimental and scientific grounding to the subject. It is now accepted by many in the machine vision community that a more rigorous approach to studying the performance characteristics of vision algorithms is required [3]. This has led in recent years to a number of special workshops and journal issues on the topic [1, 2].

Some algorithms lend themselves to direct analytic evaluation techniques such as covariance estimation and error propagation [4]. Linear approximations around operating points provide an insight into performance characteristics. However, such differential propagation techniques are inappropriate when the input errors are large compared to the range of linearity of the function or when the input distribution is non-Gaussian. Under such conditions empirical evaluation techniques such as Monte-Carlo simulations may be more appropriate. These take values from the expected input distribution and accumulate the statistical response of the output over repeated trials. There are now many good examples of both theoretical and empirical studies which highlight otherwise hidden properties of various algorithms [5, 6, 7, 8].

As vision techniques become more widely applied, the need to critically evaluate new methods has also become recognised by users. In the area of biometrics, the wide range of fingerprint, face and hand recognition systems has led to an effort to identify best practice for testing [9] and which may be useful more widely. As we shall see, there are several forms of performance characterisation. All of these methods are inherently statistical, yet there is another closely related role that statistics can play in machine vision research.

Vision algorithms must deliver information allowing practical decisions regarding interpretation of the data present in an image. One can reasonably assert that probability is the only self-consistent computational framework for data analysis, and so probability theory must form the basis of all algorithmic analysis processes. The most direct form of information regarding a hypothesis is the posterior (often conditional) probability. There are several common models for statistical data analysis, all of which can be related at some stage to the principle of maximum likelihood. The most effective and robust algorithms will be those that match most closely the statistical properties of the data. An algorithm which takes correct account of all of the data will yield an optimal result.

In many practical situations problems cannot be easily 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 which will determine the success or failure of a particular algorithm. Therefore, understanding these assumptions and compromises is an important part of algorithmic development. We can conclude that algorithms which will work best on a particular application will be those which 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.

One of the major criticisms of computer vision over the past few years has been due to a general lack algorithmic reliability. In our opinion, this has largely been due to the neglect of the important role that statistics must play in algorithm development. One could state that computer vision should strictly be regarded as a branch of applied statistics. This paper describes the dual role that statistics plays in both the evaluation and the design of algorithms.

2 Good Practice in Algorithmic Testing

In this section we discuss the issues involved in testing to obtain the performance characteristics. Running algorithms on example data sets is commonplace, and some elements of good practice may be identified.

It is important to understand that there are several forms of performance characterisation, each with its own particular rationale and benefits:

- **technology evaluation** concentrates on understanding the behaviour of specific algorithms designed to do similar tasks. The testing can be carried out off line using standardised data. The results are therefore repeatable and depend on the size and scope of the test data set. The results are performance characteristics which may be presented in terms of output parameters, their variations and density functions.

- **scenario evaluation** defines a particular use application of specific algorithms within a prototype system potentially including other components which may be more or less well characterised. The input data is based on a controlled real world and is therefore only partly reproducible. The results are cited in terms of metrics that a system user would understand such as reliability, accuracy and precision.

- **operational evaluation** concerns the utilisation of a complete system for a specific task for an end user. It may be used as a benchmark to decide whether or not a system meets a certain requirement. The evaluation must be live and on line and is therefore not precisely repeatable. Small changes in the conditions and context of use may have quite dramatic effects on the system performance, especially when human users are involved.

2.1 Test Objectives

The objective of a technology evaluation is to understand the performance characteristics of an algorithm. This will consist of a series of tests, each of which will have a specific objective. It is important to carefully define the objective of each task, since even small differences will make it change the form of the test and make it hard to compare similar tests.

2.2 Test Data

A typology of test data has been proposed [10] and includes:

- Simulations using data without noise, allowing verifications of performance. This corresponds to the study in [7] which revealed the stability conditions.
- Simulations using data with noise: perhaps the most common form
- Empirical testing using real data with full control (technology evaluation)
- Empirical testing using partially controlled test data (scenario evaluation)
- Empirical testing in an uncontrolled environment (operational evaluation)

Note that these map well onto the types of evaluation described above.

Test data consists of three elements (1) the test data, (2) the corresponding ground truth and (3) other information about the data such as its origin and conditions of capture. Ground truth is an estimate of what is thought to be in the test data. It may be determined by an independent method, or it may be manually defined. Manual annotation provides a partial ground truth in that it is somewhat subjective. In fact unless the data is synthetically generated, ground truth will always have some residual error rate (bias and imprecision) due to administrative or instrumental error.

Clark and Courtney reviewed issues of data set design and selection [11] for optical character recognition (OCR), motion estimation and face recognition. They concluded that the OCR field is perhaps the most well developed in terms of data set collection. It is perhaps no coincidence that this is also the area for which it is easiest to obtain ground truth. Even then, a residual error rate of better than 50 parts per million character substitution error proves very hard to achieve.

The amount of data is a critical issue and there has been some work in this area [12]. Robust algorithms may require huge quantities of data. An algorithm with 99% reliability means an error rate of 1%, which in turn means that hundreds of test images may be required. It is worth noting that algorithms which provide a covariance matrix or confidence on the results will require less test data than one which just produces a result. In this case all that is required is to demonstrate that the prediction of algorithm accuracy is within some margin (which could be as large as a factor of two) of the actual estimates. An algorithm which is capable of estimating its own accuracy is then relatively easily incorporated into a system. Such an approach should even be capable of identifying situations where insufficient (or too strongly correlated) data is provided for accurate determination of the required result.

The range of data to be collected is also important. It should neither be too easy not too hard and should be representative of the range of conditions. One class of conditions is that which is not expected to affect the performance of an algorithm, to ensure **invariance** to this change. For example paper deformation for a scanned document [13]. This class of data can often be generated synthetically from real data and forms a hybrid in the typology described earlier. It has the potential to vastly increase the effective size of the test data set. Test data is often costly to collect. OCR data was estimated to cost between $15 and $25 per page when the ground truth was entered 4 ways to minimise the error rate. The high cost of quality data is an argument for sharing data whenever possible.

2.3 Test Metrics

In order to carry out a test, it is necessary to define a metric that can be used to quantify performance. The metric should be **quantitative** (to permit comparison between algorithms) and **objective** (that is unbiased and repeatable). Such performance metrics can usefully be associated with the failure modes of an algorithm. The failure mode will depend on the function of an algorithm. We can begin by considering the case of a feature detector, and realising that one failure mode is the inability to detect.

2.3.1 Feature Detection Reliability

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.

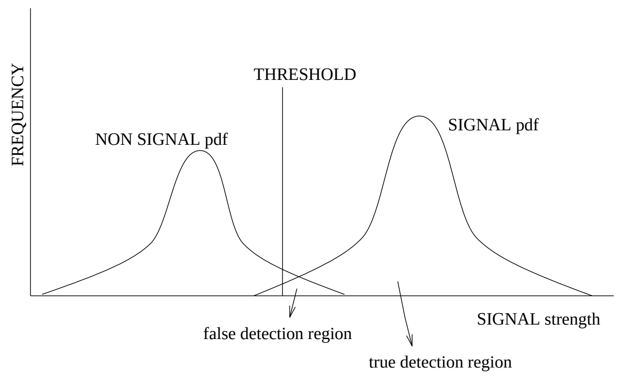


Figure 2: signal and non-signal detection pdfs

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.

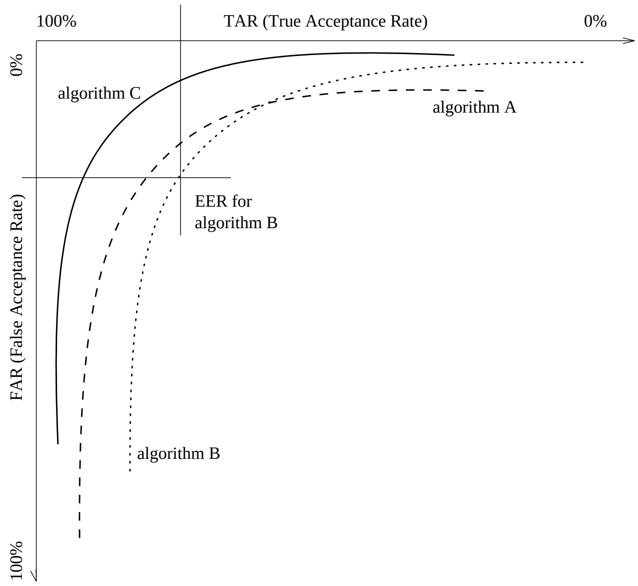


Figure 3: Receiver Operator Curve

In the ROC curve [1], one axis represents the True Acceptance Rate (TAR) and the other represents the False Acceptance Rate (FAR) [2]. 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 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

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

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

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 if any reasonable level of true detection is to be expected. We will pick up on this point again later on.

2.3.2 Other Metrics

Other types of algorithms have their own appropriate performance metrics according to the function they perform. Although a definitive list has yet to appear, it might include the following:

- **feature localisation** may be described in terms of the parameters of that feature, including location accuracy and bias. In the case of edges this might include orientation [14].
- **matching algorithms** such as those employed in stereo or motion estimation may be specified in terms of true and false matches [15]. The lack of a gold standard can be finessed by appealing to reproducibility [16].
- **parameter estimation** problems such as motion estimation, registration and calibration estimation, should present results together with a covariance on the parameters [3].
- **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.

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.

A problem arises if the task involves **both recognition and localisation**, or involves **multiple or parameterised objects**. This is the situation in document image analysis, where graphical symbols and regions must be recognised and where there exists potential ambiguity in for example the classification of a feature as a long dashed line or a series of short colinear lines. Metrics defined in this area (see [17]) typically involve some measure of overlap and some tradeoff in the mis-recognition of simple and complex objects.

- **segmentation**: A review of segmentation evaluation techniques [18] identified two classes of empirical evaluation: *goodness* methods that attempt to quantify the quality of the output image based on measures of region shape, uniformity and contrast; and *discrepancy* methods that compare the segmentation results with some ground truth based on measures of pixel misclassification (number or location) and number of regions. Studies in [19] utilise measures of true and false regions with a specified overlap compared to a manual segmentation as a way of determining the degree of over and under segmentation. For **unsupervised classification** only *goodness* methods seem applicable, but these are unsatisfactory (biased) if they correspond to the optimisation criteria used in the segmentation algorithm being evaluated. Other metrics have been suggested [20].
- **closed loop systems**: Little work has been carried out into the performance of such systems (but see [21]) though systems of this kind may be able draw upon stability parameters from control theory.

2.4 Test Protocols

The apparent performance of an algorithm in a test will depend on the test data used and, in many cases, the training data used to train the algorithm. It is hoped that the training data will be sufficiently large so as to be fully representative of the test data, but in reality a finite amount of data is available and must be partitioned into training and testing sets. In order to control generalisation of the training data and to avoid overlearning, the training data itself may be partitioned into training and validation sets. Several test protocols have been proposed (see [22] for a recent review):

- The **leave all in** approach involves using all the available data for training and all the data again for testing. This ensures that the training is as complete as possible with the available data. However, it will produce an optimistic view of performance since there is total overlap between the training and test data which is unrealistic. This technique is not able to detect an algorithm with poor generalisation ability.
- **cross validation** methods involve keeping part of the data for training and using the rest for testing. There are several variants:
  - **holdout techniques** suggest testing with a fixed fraction of the data. The estimate produced is pessimistic in that there is strictly no overlap between the training and test data, which again is unrealistic.
  - The **leave one out** method suggests training with all the data except one item, and testing with that item. This will be more representative than the holdout, but

the metric will suffer from high variance due to the chance of selecting an easy or difficult test item.

  - **rotation** involves repeatedly generating training and test sets by random partitions of the available data. This allows a picture to be built up of average performance and the confidence intervals for the metrics used. The problem with this technique is that it can require an inordinate amount of time, since they use all the data and an entire training cycle for each test.

- **resampling** methods involve generating a training set and a test set by resampling the entire data set at random. This gives training and test sets which are of arbitrary size and may contain no, one, or more than one instance of a given data item. Variants include the bootstrap and the jack knife [23]. These are cheaper than the leave one out technique with rotation, and appear to provide better error rate estimates than the leave all in and holdout techniques. They are especially useful when only small data sets are available.

It has been suggested that for limited amounts of test data, an unbiased estimate of performance may be obtained by taking the average of the leave all in and leave one out estimates [24] on the basis that the the error measure for the two protocols converge at infinite data.

2.5 Test Administration

Another important issue is that of how testing is organised. Whilst it is perfectly feasible for an individual group to define and perform a test, there are a number of alternatives:

- **Testing by an independent body**: Examples include the testing of **optical character recognition system** for use in the US census [25, 26] and the FERET tests of **face recognition** algorithms by the US Army [27]. The involvement of developers at workshops and open feedback played an important role in encouraging the improvement of the technology.

- **Collaborative testing**

  - A test of **image registration** of medical image sets was reported in [28]. Metrics for defining registration accuracy were agreed, and 14 pairs of image set were provided to 12 sites for processing by 16 techniques, the results of which were collated by a central site. Whilst some methods gave lower median errors than others, the methods were not required to produce covariance matrices and so were not able to predict their own accuracy. The small number of data sets used did not make it possible to determine the most suitable methods for a given task.
  - A test of **range image segmentation** techniques [19] was performed on a set of range and reflection images. Four groups compared their algorithms on a set of 30 training images and 10 (manual) ground truth test images from two scanners. The data and scoring code have been made available [30].
  - A series of tests on **stereo matching** was carried out involving 15 groups [29] and similar types of tests have been organised for graphics recognition. In each case the developers are presented with a set of images and given a certain amount of time to process the data with their algorithms.

- **Closed competitive** tests are run by developers demonstrating their own techniques. They may use commonly available techniques as their baseline, or reimplementations of a state of the art algorithm. The later may be quite problematic, as it is hard to ensure that the implementations are correct and this may detract from the credibility of the tests. One solution is to make available open source versions of code such as the Barron and Fleet optical flow code [31] and the TINA software package [32]

Some other examples are listed on the ECVNet website [33].

Although performance metrics and protocols can be agreed, they may have a wide range of interpretations which means that it is very easy to generate results that are not directly comparable. For example accuracy may be defined as absolute or relative, precision in terms of the number of standard deviations or percentage confidence intervals. One solution is to prepare a software package that automatically and objectively calculate the metrics and prepares the required tables and plots. This would include the task specification, data set with annotation or ground truth, the test metric scoring code and baseline algorithm. Such a package can then be made available on the internet as in [30].

3 Algorithmic Design Principles

We turn our attention now to the role that statistical methodology can play in algorithm design. In this final section we discuss aspects of good algorithmic design including understanding of the statistical properties of data and common algorithmic operations. We show how the method of error propagation can be used to assess the stability of image processing algorithms, and how knowledge of the origins of the method of least squares can lead to evaluation techniques and modifications which result in better algorithms. Familiarity with the common forms of the equations used in computer vision often allows the statistical assumptions regarding the data to be deduced[3]. Knowledge of the adequacy of these assumptions has a direct and beneficial effect on algorithmic effectiveness. Successful assumptions in one algorithm will generally be of use in other algorithms, allowing rapid algorithmic development for new problems.

3.1 Property of Common Image Arithmetic Operations

In simple image processing, the requirements of an image algorithm may be purely to enhance the image for viewing. However, the aim of image operations for image analysis and computer vision is to produce an image that make certain information explicit in the resulting image to assist in automated data extraction. Generally, final stages of the algorithm result in high values over features of interest which are then thresholded to define a set of features or regions. Leaving aside the definition of the feature, one criteria which determines a good feature detection algorithm is its behaviour in the presence of noise. In particular, that the resulting image gives results that really can be interpreted purely on the basis of the output values. This ensures that a high value is likely to be genuine and not just a consequence of propagated noise. Only stable techniques can form the basis of the useful building blocks for image analysis algorithms.

[3] clearly, good papers will already have identified and validated the underlying assumptions

It is therefore useful to examine the properties of some common image operations using error propagation. Following the general approach for error propagation, which assumes error distributions which are well characterised by variances which are small in comparison to the typical range of linearity of the processing functions, we recall that:

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

where C_X is the covariance on the measured values and ∇f is a vector of derivatives:

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

and $\Delta f^2(X)$ is the variance on the computed measure. If we apply this to image processing operations, and assume that images have uniform independent random noise, then we can simplify this expression to:

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

That is, the contribution to the output from each variance involved in the calculation is added in quadrature and weighted with the linearised approximation to function stability. Applying this analysis to a set of common image operations for input image I, output image O and output image variance ΔO^2. we get the following results:

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. Error Propagation in Image Processing Operations.

Notice that for only one of the operators (addition) is the output variance independent of the image data. This algorithm preserves uniform random noise in the output image. Many of the very common image arithmetic operations, most notably multiplication and division, produce a result which is **data dependent**. This means that each output pixel will have different noise characteristics. So the noise level in one part of the image will be different from that in another part of the image, according to the image content. This complicates the process of interpretation since it makes it much harder to set a threshold which has any statistical meaning.

On the whole, more complicated functions will result in more complicated noise characteristics and this gives another theoretical motivation for Bowyer's conjecture. However, this is not always the case. For example: Square-root of Sum of Squares:

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

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

This calculation contains several arithmetic components which we have already shown to be unstable. Yet in the case of equal variances $\sigma_1^2 = \sigma_2^2$ this reduces to the stable form:

$$\Delta O^2 = \sigma^2$$

Such results give us the motivation to continue investigating sophisticated approaches to data analysis.

3.2 Linear Filters

For linear filters we introduce the spatial subscript for the input and output images I and O:

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

where h_{nm} are the linear co-efficients. Error propagation gives:

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

For uniform errors this can be rewritten as:

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

So that for example a simple filter with:

$$h = (-1, 0, 1)$$

This would give:

$$\Delta O^2 = 2\sigma^2$$

Thus linear filters produce outputs that have uniform errors. However, whilst these errors have the desirable property of being uniform, they are no longer independent, unlike those from image arithmetic. This is because the same data is used several times in the calculation of the output image pixels. So care must be taken when applying further processing.

If further linear filtering stages are used, the principle of superposition means that all the operations can be replaced by a combined linear filter and the original derivation still holds.

3.3 Combining Image Operations

Algorithm design has conventionally progressed by developing and applying a number of data manipulation processes that attempt to extract some set of values from an image. The Central Limit Theorem shows that as multiple noisy processes are combined, they will tend towards a Gaussian distribution. Whilst this may be so in the limiting case, one cannot rely on this when dealing with small numbers of poorly behaved distributions. This is especially so in non-linear processes such as image processing steps since even very small deviations from the Gaussian distribution on the input data can have catastrophic effects on algorithm performance and thus on the output data. As modules are chained together, they produce a set of partial results which may become more and more non-linear and have increasingly non-uniform error properties as one moves down the chain. The final output of such a system is likely to be data that is fragile, fragmented and unstable. The combined effects of data dependent calculations and local correlations can mean that the results of feature detection algorithms often visually resemble maggots squirming about on the screen. Saying, that the results have "gone to maggots" would not be a bad analogy.

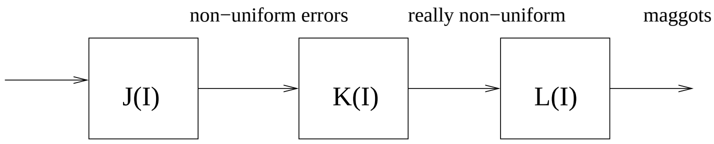


Figure 4: Propagation of errors though an unstable algorithm

An alternative approach to algorithmic development would be to try to maintain the distribution of errors such that they are of some known and manageable form. An example might be uniform errors.

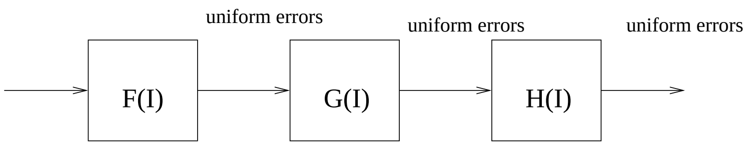


Figure 5: Propagation of errors though a stable algorithm

Edge detection is an example of one such processing chain which comprises a sequence of processing steps:

- a convolution with a linear noise filter: $\otimes$ Gaussian
- a calculation of spatial derivatives ∇_x, ∇_y with local differences: $\otimes$ (-1, 0, 1)
- a calculation of edge strength: $\sqrt(\nabla_x^2 \;+\; \nabla_y^2)$

This is followed by a thresholding and peak finding stage. This final stage will be reliable provided that we have stability after the first three image processing steps. From our analysis of the basic elements, filtering and sum of squares appear to be stable and one can conclude that edge detection is a stable operation.

The Canny edge detector [34] follows this sequences of steps. The success and wide use of this operator is often though to be due to the application of optimality criteria in its design. In reality the Gaussian smoothing used is sub-optimal and the success of the operator may have more to do with the error preserving properties of the constituent steps.

However, it should now be clear that such stability may not be exhibited by other image extraction operations. Such arguments may explain why it has proven difficult to design reliable corner detectors.

3.4 Maximum Likelihood

We turn now to the problem of interpretation of data using a model. We start with some basic definitions of probability

- $P(A)$ probability of event A.
- $P(AB)$ probability of simultaneous events A and B.
- $P(A|B)$ probability of event A given event B.
- $P(A|B,C)$ probability of event A given events B and C.

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

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

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

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

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

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

which allows us to compute the probability of one event in terms of observations of another and knowledge of joint distributions. We can extend the joint probability equation to three or more events:

$$P(A,B,C) = P(A|B,C)P(B,C)$$

$$P(A,B,C) = P(A|B,C)P(B|C)P(C)$$

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

$$P(X_0, X_1, X_2 ... X_n|\Theta)P(\Theta) =$$

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

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

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

$$P(\mathbf{X}|\Theta) = P(X_0|\Theta)P(X_1|\Theta)P(X_2|\Theta)...P(X_n|\Theta)$$

Clearly this is a more practical definition of joint probability. However, it does 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.

If we define the variation of the observed measurements X_i about the generating function (at Y_i) with some random error, the probability:

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

will be equivalent to $P(X_0|\Theta, Y_0)$. As selection of data point X_0 automatically specifies Y_0 this latter term can safely be dropped. Choosing Gaussian random errors with a standard deviation of σ_i gives:

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

where A_i is a normalization constant. This may be used as the basis for the maximum likelihood which is derived from the joint probability:

$$P(\mathbf{X}, \Theta) \;=\; (\Pi_i P(X_i|\Theta))P(\Theta)$$

We can now construct the maximum likelihood function:

$$log(P(\mathbf{X}|\Theta)) \;=\; \sum_i log(P(X_i|\Theta))$$

such that:

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

which leads to the χ^2 definition of log likelihood:

$$log(\mathbf{X}|\Theta) = -\frac{1}{2}\sum_i \frac{(X_i - f(\Theta, Y_i))^2}{\sigma_i^2} + const$$

This expression can be maximized as a function of the parameters Θ and this process is generally known as a least squares fit. The importance of this to machine vision is that the majority of all algorithms can be traced back to this single estimation technique. In particular, any solution of sets of linear equations by matrix inverse, many neural network training algorithms, and generally most optimality criteria. However, there are several key failings of such an approach when used as the basis for machine vision algorithms. It is based on the assumptions of independent measurements and Gaussian errors. This can break down in a number of ways:

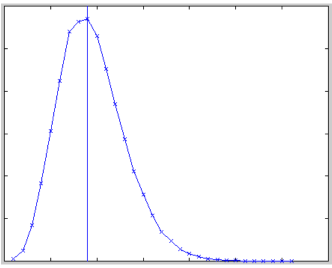
- non-Gaussian data
- non-independence in the measurement parameter space
- non-linearity in the measurement parameter space

Much effort is directed, sometimes unknowingly, into overcoming these limitations. What follows below are simple suggestions for investigative plots which can be generated to test the adequacy of these assumptions and suggestions for modifications to the basic approach should poorly conforming data be identified.

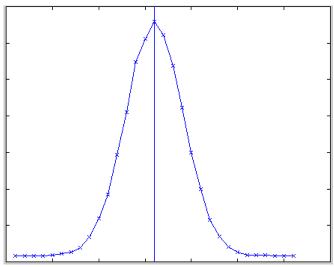
3.4.1 Non-Gaussian Errors

Whereas least squares implicitly assumes a Gaussian distribution in the measurement variable, machine vision is full of data for which this cannot be assumed. There are two forms of the problem:

- The error distribution may be relatively compact but badly skewed
- There may be outliers caused by data "contamination".



(a) skew



(b) outliers

Figure 6: data distributions featuring skew and outliers

One technique for coping with the first problem is to transform the data to remove skewing. This can be done if we have a relationship between the measurement and its standard deviation:

$$\Delta_x \; = \; f(x)$$

we can seek a function g which will give us:

$$\Delta_{g(x)} \; = \; const$$

Using error propagation:

$$\Delta_{g(x)} \; = \; \Delta_x \, dg/dx \; = \; f(x) \, dg/dx \; = \; const$$

That is, integrate the reciprocal of the error dependence:

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

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

If we take stereo data as an example, we have depth z related to feature locations in the left x_l and right x_r images so:

$$z \; = \; \frac{fI}{(x_l \; - \; x_r)}$$

Errors in position $p = (x, y, z)$ are therefore badly skewed. Attempting a least mean squared solution with these measures directly (eg for model location) is unstable due to large errors for large z. However, errors in **disparity space** $d = (x, y, 1/(\sqrt{2}z)$ are uniform and can be used for fitting.

3.4.2 Dealing with Outliers

As discussed earlier under the performance of detection algorithms, there is a trade off between true and false detection, such that for any reasonable level of reliability (true detection rate) there will always be some rate of false signal. This will manifest itself as outliers in the data. So any algorithm processing features will have to deal with outliers. This problem can be recognised in any practical problem by simply producing a histogram of the residuals from fitted values. What to do about such outliers once the problem has been identified is a fundamental research issue.

The area of **Robust Statistics** offers a number of techniques which can help [35]. The simplest involves limiting the contribution of any data point to the total least mean square:

$$-log(P) \;=\; \sum_i min(\frac{(X_i - f(\Theta, Y_i))^2}{\sigma_i^2}, 9.0)$$

The choice of 9.0 (3 σ) as the limit on the contribution is approximate and may be adapted to the problem.

This technique is not particularly good for methods which use derivatives during optimisation, as it introduces discontinuities that may lead to local minima. An alternative involves replacing the Gaussian with a continuous distribution with long tails. The most common of these is the double sided exponential which is adequate for most applications:

$$-log(P) \;=\; \sum_i |\frac{(X_i - f(\Theta, Y_i))}{\sigma_i}|$$

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

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

so that our log probability is now:

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

These are continuous, so we can use derivative methods for optimisation. However, the price we pay is that, unlike standard least squares, such cost functions can rarely (or perhaps never) be optimised by **direct** solution so we have to use **iterative** techniques which tend to be slower and can also result in local rather than global optima. Interestingly, a link between the highly robust technique of the Hough transform and maximum likelihood has been shown in [36] and this provides one method of locating global minima using a robust statistic.

3.4.3 Non-Independent Measurements

Under practical circumstances the data delivered to an algorithm may be correlated. Correlation produces systematic changes in one parameter due to changes in another. This can be visualised by producing a scatter-plot of the two variables $f(x, y)$. In general for any two variables to be un-correlated knowledge of one must give no information regarding the other.

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

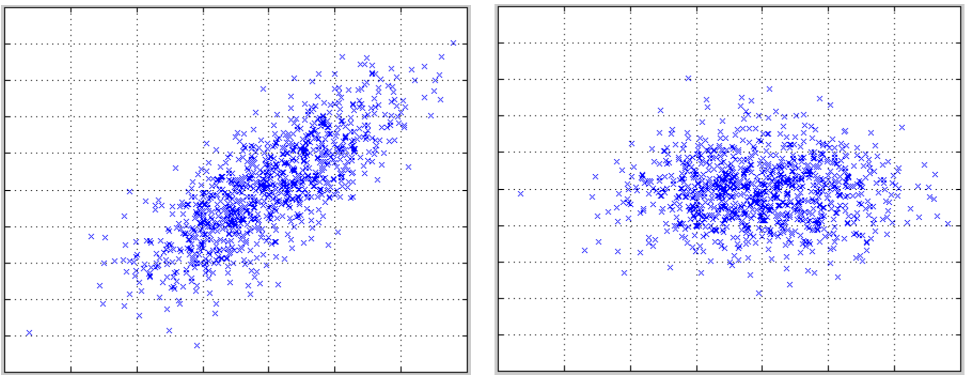
$$f(x, y) \ = \ f(x) \otimes f(y)$$

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:

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

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



(a) original (b) rotated

Figure 7: original and rotated data distributions

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) approximates a matrix by a set of orthogonal vectors and singular values, and it can be shown that the singular vectors satisfy the eigenvector equation with:

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

3.4.4 Identifying Non-linear Correlations

Whilst principal component analysis works by rotating the axes of the space to align along the axes of major variance of the data, this may not necessarily de-correlate the data. In particular if there are non-linear relationships within the data no amount of rotation will achieve this.

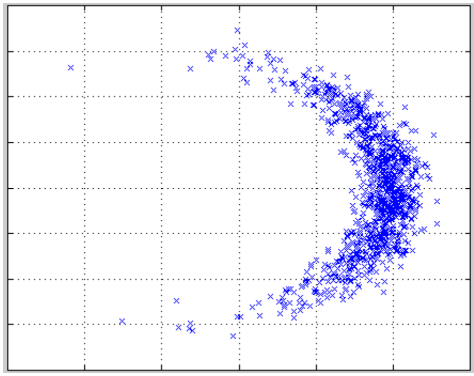


Figure 8: non linear data distribution

There exist techniques for decorrelating non-linear relationships. These include bottleneck neural networks [37], kernel PCA [38], and support vector machines [39], but these methods are often quite difficult to use.

4 Conclusions

4.1 Rigorous Testing is Feasible

Unless algorithms are evaluated in a manner that can be used to predict the capabilities of a technique on an arbitrary data set, it is unlikely to be successfully reimplemented and used. The subject cannot advance without a scientific methodology, which it will not have without an acknowledged system for evaluation, characterisation and the ability to reimplement (or at least share) algorithms. We maintain that algorithms which cannot be tested will **never** be used in anger. Rigorous approaches to testing exist, and we have listed here some of the main evaluation criteria for common groups of algorithm, but the main obstacle may be largely psychological.

4.2 Algorithmic Design Principles Produce Better Algorithms

We have shown that the properties of common image operator are not all the same and that some are more stable and thus safer to use than others. The assumptions underlying maximum likelihood have been given and some techniques presented for ensuring that the data to be processed will conform to the constraints imposed by these assumptions. Robust Statistics will probably be needed for all practical problems, though covariances can still be computed. Error propagation can be used to assess the effects of noise and guide the design of stable algorithms. Monte-Carlo techniques can be used when all other methods are unsuitable.

Design methodologies such as these make explicit the assumptions embedded in an algorithm and therefore go to the heart of problems arising due to Bowyer's conjecture. Explicit evaluation of each algorithmic assumption gives us at least the possibility of controlling these effects. On the other hand we have also demonstrated that problems of algorithmic stability can give rise to additional sources of unreliability. In these cases perhaps only simple techniques or very carfully designed algorithms will allow us to avoid the undesirable consequences of Bowyer's conjecture. If this is the case then, contrary to the popular trend, researchers should avoid undue sophistication in algorithms until it is warrented by the characteristics of the data.

Acknowledgements

The authors wish to thank Ross Beveridge at Colorado State University, Kevin Bowyer at the University of South Florida, Henrik Christensen at KTH, Adrian Clark at the University of Essex, Wolfgang Foerstner at the Institute for Photogrammetry in Bonn, Robert Haralick at the University of Washington and Jonathon Phillips at NIST for their contributions to the area and whose work has inspired this review. The support of the Information Society Technologies programme of the European Commission is gratefully acknowledged under the PCCV project (Performance Characterisation of Computer Vision Techniques) IST-1999-14159.

References

[1] K.W. BOWYER AND P.J. PHILLIPS, *Overview of Work in Empirical Evaluation of Computer Vision Algorithms*, IN K.W. BOWYER AND P.J. PHILLIPS, *Empirical Evaluation Techniques in Computer Vision*, IEEE Computer Press, 1998.

[2] 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.

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

[4] R. M. HARALICK, *Covariance Propagation in Computer Vision*, Proceedings of ECCV Workshop on Performance Characteristics of Vision Algorithms, Cambridge, UK, April 1996.

[5] 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.

[6] N. GEORGIS, M. PETROU AND J. KITTLER, *Error Guided Design of a 3D Vision System*, IEEE Trans PAMI, 20(4), 1998, pp.366-379.

[7] 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.

[8] P. COURTNEY, N.A. THACKER AND A.F. CLARK, *Algorithmic Modeling for Performance Evaluation*, Machine Vision Applications, 9 (5/6), 1997, pp.219-228.

[9] BIOMETRIC WORKING GROUP, *Best Practice in Testing and Reporting Performance of Biometric Devices version 1.0*, Biometrics Working Group, 12 Jan 2000, www.afb.org.uk/bwg/bestprac.html

[10] W.M. MAIMONE AND S.A. SHAFER, *A Taxonomy for Stereo Computer Vision Experiments*, Proceedings of ECCV Workshop on Performance Characteristics of Vision Algorithms, Cambridge, UK, April 1996.

[11] A.F. CLARK AND P. COURTNEY, *Databases for Performance Characterization*, IN R. KLETTE, H.H. STIEHL, M.A. VIERGEVER AND K.L. VINCKEN, *Performance Characterization in Computer Vision*, Kluwer series on Computational Imaging and Vision, 2000.

[12] I. GUYON, J. MAKHOUL, R. SCHWARTZ, AND V. VAPNIK, *What Size Test Set Gives Good Error Rate Estimates?*, IEEE Trans PAMI, 20(1), 1998, pp.52-64.

[13] T.M. HA AND H. BUNKE, *Off-line Handwritten Numeral Recognition by Perturbation Method*, IEEE Trans PAMI, 19(5), 1997, pp.535-539.

[14] E.P. LYVERS AND O.R. MITCHELL, *Precision Edge Contrast and Orientation Estimation*, IEEE Trans PAMI, 10(6), November 1988, pp.927-937.

[15] N.A. THACKER AND P. COURTNEY, *Statistical Analysis of a Stereo Matching Algorithm*, British Machine Vision Conference, BMVC92, Leeds, UK, pp. 316-326, September 1992.

[16] Y.G. LECLERC, Q.T. LUONG AND P. FUA, *Measuring the Self-Consistency of Stereo Algorithms*, Proceedings European Conference on Computer Vision ECCV 2000, Dublin, Ireland, pp.282-298, June 2000.

[17] I.T. PHILLIPS AND A.K. CHHABRA, *Empirical Performance Evaluation of Graphics Recognition Systems*, IEEE Trans PAMI, 21(9), 1999, pp.849-870.

[18] Y.J. ZHANG, *A Survey on Evaluation Methods for Image Segmentation*, Pattern Recognition, 29(8), 1996, pp.1335-1346.

[19] A.W. HOOVER, J. JEAN-BATISTE, X. JIANG, P. FLYNN, H. BUNKE, D. GOLDGOF, K.W. BOWYER, D. EGGERT, A. FITZGIBBON AND R.B. FISHER, *An Experimental Comparison of Range Image Segmentation Algorithms*, IEEE Trans PAMI, 18(7), 1996, pp.673-689.

[20] 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.

[21] P. COURTNEY AND J.T. LAPRESTE, *Performance Evaluation of a 3D Tracking System for Space Applications*, DAGM Workshop on Performance Characteristics and Quality of Computer Vision Algorithms, Braunschwieg, Germany, 18 Sept. 1997

[22] A.K. JAIN, R.P.W. DIUN AND J. MOA, *Statistical Pattern Recognition: A Review*, IEEE Trans PAMI, 22(1), 2000, pp.4-37.

[23] B. MATEI, P. MEER AND D. TYLER, *Performance Assessment by Resampling: Rigid Motion Estimators*, IN K.W. BOWYER AND P.J. PHILLIPS, *Empirical Evaluation Techniques in Computer Vision*, IEEE Computer Press, 1998.

[24] J. PIPER, *Variability and Bias in Experimentally Measured Classifier Error Rates*, Pattern Recognition Letters, 13, pp.685-692, 1992.

[25] J.L. BLUE, G.T. CANDELA, P.J. GROTHER, R. CHELLAPPA AND C.L. WILSON, *Evaluation of Pattern Classifiers*, Pattern Recognition, 27(4), 1994, pp.485-501.

[26] R.A. WILKINSON, J. GEIST, S. JANET, C.J.C. BURGES, R. CREECY, B. HAMMOND, J.J. HULL, N.J. LARSEN, T.P. VOGL AND C.L. WILSON, THE FIRST OCR SYSTEMS CONFERENCE, Technical Report NISTIR 4912, NIST, 1992. THE SECOND CENSUS OCR SYSTEMS CONFERENCE, Technical Report NISTIR 5452, NIST, 1994.

[27] P.J. PHILLIPS, H. MOON, S.A. RIZVI AND P.J. RAUSS, *The FERET Evaluation Methodology for Face-Recognition Algorithms*, IEEE Trans PAMI, 2000.

[28] J. WEST, J.M. FITZPATRICK, ET AL, *Comparison and Evaluation of Retrospective Intermodality Brain Image Registration Techniques*, J. Comput. Assist. Tomography, 21, 1997, pp.554-566.

[29] E. GUELCH, *Results of Tests on Image Matching of ISPRS III/4*, Intl. Archives of Photogrammetry and Remote Sensing, 27(III), 1988, pp.254-271.

[30] www.marathon.csee.usf.edu/range/seg-comp/SegComp.html

[31] BARRON AND FLEET, *optical flow code and data*, ftp.csd.uwo.ca/pub/vision. See also J.L. BARRON, D.J. FLEET AND S.S. BEAUCHEMIN, *Performance of Optical Flow Techniques*, Intl. J. Computer Vision, 12(1), 1994, pp.43-77.

[32] *TINA*, www.niac.man.ac.uk/TINA

[33] *ECVNet Benchmarking and Performance Evaluation Website*, pandora.imag.fr/ECVNet/Benchmarking.html

[34] J. CANNY *A Computational Approach to Edge Detection*, IEEE Trans PAMI, 8(6), 1986, pp.679-698.

[35] P. MEER, P. MINTZ, A. ROSENFELD AND KIM, *Robust Regression Methods for Computer Vision: A Review*, Intl. J. Computer. Vision, 1991.

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

[37] P.D. SOZOU, T.F COOTES, C.J. TAYLOR AND E.C. DI MAURO, *Non-linear Point Distribution Modeling using a Multi-layer Perceptron*, British Machine Vision Conference, BMVC95, Birmingham, UK, September 1995.

[38] www.kernel-machines.org

[39] V.N. VAPNIK, *Statistical Learning Theory*, John Wiley, 1998.