Stable Segmentation of 2D Curves

Andrew W. J. Fitzgibbon

Ph. D.
University of Edinburgh
1997

Contents

Chapter 1

Introduction

A fundamental issue in computer vision is the choice of representation of shape. It is the choice which motivates and dictates the selection of techniques for recovery, analysis and ultimately interpretation of the visual scene. Throughout the history of machine vision research, one of the primary themes has been the development and exploration of many schemes which attempt to encode the shape of objects in the world efficiently and robustly. While shape representation has seen many forms, this thesis concentrates on two-dimensional planar curves and in particular on the tradeoffs inherent in the choice of representation. This chapter provides the motivation for the work of the thesis and outlines the methodology within which the explorations herein take place. Finally the contributions of the research are briefly presented.

1.1 Background: The canonical vision system

We are interested in recognising complex curved objects. Typical scenes contain one or many objects, possibly overlapping. We make the assumption that the objects in the scene are a subset of a database of known objects stored in the computer, and that novel objects will be indicated to the system by an external agent. Although there are several broad strategies for object recognition, the framework in which most recognition systems work is a model-based one: the known objects are stored as geometric models and the image is processed in order to recover these models [13, 18, 81, 44, 72, 32, 4, 15, 47, 54, 34, 31, 37, 45, 39, 122, 112, 136]. Given this framework, a number of decisions remain to be made, choice of model representation and matching algorithms being most important. The data representation is largely predetermined, a 2 or $2\frac{1}{2}$D image, and systems are conveniently divided into three functional parts:

1. **Segmentation** converts the input image into a form suitable for matching against the chosen model representation, by a combination of data partitioning and feature fitting. This is the area in which the work reported here contributes. The combination of model representation and segmentation output must be sufficient to allow the system to:

2. **Invoke** or index [33, 64, 122] a small number of plausible hypotheses of data-to-model pairings from the (possibly large) object database. While in industrial applications, the number of objects can generally be controlled, this is not the case in less structured environments. It seems desirable that a system should be extendible to cope with such situations.

3. Establish **correspondences** [45] between features in the model and points in the image. This is of additional importance in the industrial domain where inspection

tasks are specified in terms of measurements to be made on or between labelled model features.

Within this framework, the problem of establishing correspondences between model and data features (the so-called *correspondence problem*) is often considered to be the bottleneck process. However, this process is greatly simplified if a stable and powerful image segmentation is available. If a scene containing a single rigid object is converted to a symbolic form which is exactly that chosen in the model, modulo a predefined class of allowed transformations, matching will be exact.

1.2 This thesis: Curve segmentation

This thesis deals with the specific subproblem of converting sequences of two-dimensional data points into higher level descriptions in terms of curved shape primitives. This is an important member of the wider family of segmentation problems, as raw curve data are readily available from edge-detection processes, but higher level processes—such as the extraction of geometric invariants or 2D object recognition—require the higher level descriptions. The overriding principle of the work is that if the segmentation process is accurate and repeatable, the later stages of processing are greatly simplified. In order to present the goals of the thesis, I first introduce two concepts that will facilitate a more precise description of the problem and the contributions of this work: the Marr-Brady criteria for representation of shape, and the adaptation of Marr's computational theory of vision to the specific problem of curve segmentation.

1.2.1 Background: Criteria for representation of shape

As the next chapter shows, many proposals have been made for shape representations, and, despite the work of Marr [73] and Brady [17], selection among these alternatives is still largely based on intuition, received wisdom and the personal experience of researchers.

Formalism of these choices was eloquently discussed first in [74], and later in [17]. Marr and Nishihara's paper introduced the criteria by which they proposed to evaluate potential shape description schemes:

- The **scope** or *power* of the representation describes the range of objects that can be represented. A representation should ideally allow the representation of any arbitrary object, but in practice it is sufficient to remain limited to objects in the domain of activity of the vision system.

- The representation of any object should be intrinsically *invariant* to shape-preserving transformations. Marr calls this the **uniqueness** criterion. Invariance is the property that the segmentation, expressed in the object's reference frame, is invariant to a predefined class of transformations of the scene and image. In 2D recognition systems, we are normally concerned with invariance to affine and projective transformations of the image plane, while in 3D systems, the required invariance is frequently rigid-body transformations of the imaged objects.

  An effective segmentation system will be at least theoretically viewpoint invariant, although practical constraints may limit this invariance to smaller variations in appearance. For example, although the ellipse fitting algorithm of Chapter 3 is theoretically invariant to affine transformations of the image points, large affine skews or changes in scale will render it sensitive to quantization effects in the image.

- **Stability** encodes the additional important requirement of (near) invariance to noise and occlusion while possibly conflicting with the argument for **sensitivity**: the ability to distinguish subtle changes in the underlying shape.

- Finally, they required that the representation be **accessible**, derivable from the image within the time and space constraints of the vision system's application. This is also an argument for *concise descriptions*—although matching a perfect segmentation will in general be exact, we cannot conclude that it will be quick. The speed of the algorithm is dependent on the length of description—small rich descriptions will invoke and match quickly, whereas a valid, but large, description will in general take longer.

To these criteria, Brady adds several more, only two of which I will emphasize here.

- The **richness** criterion extends Marr's concept of scope to insist that the representation be as information-preserving as possible. This effectively adds an *invertibility* requirement to allow the shape to be recovered from the representation and paves the way toward this thesis' concentration on parametric curves. From parametric descriptions, the final symbolic representation of the scene should be sufficient to reconstruct the original image (modulo noise), thereby ensuring that no information has been lost in the conversion.

- **Local support** is the property that the description ought to be computable from a small local subset of the data. In arguing for local support, Brady stresses computational tractability but now it is more important to note that local support is a *sine qua non* of invariance to occlusion. Chapter 4 experimentally determines that the degree of occlusion is a primary determining factor in the performance of conic-fitting algorithms.

1.2.2 Background: A computational theory of curve segmentation

Marr's computational theory [73] provides a powerful tool for reasoning about information processing devices. He describes the computational approach as comprising three levels at which such a device must be understood. At the top level, we define the *abstract computational theory* which characterizes the mapping of information from the device input to output. The second level describes the *representation and algorithms* that will be used to perform the information transformation; while the lowest level deals with the *hardware implementation*. The key to this approach is in the separation of the task definition from the specifics of the algorithm which will be used to extract the information. There is perhaps no more compelling demonstration of the power of this methodology than the success of one of its early applications—the development of the Canny edge detector [21].

This thesis introduces such a computational definition of the segmentation problem and characterizes it as a curve fitting problem, comprising the choice of shape representation and error-of-fit metric. Not until the algorithmic level need we consider the fitting strategy. Of course (as Marr notes) without a computationally tractable and robust algorithmic level, the computational description provides little benefit to potential users of the system. In the paradigm presented here, the traditional segmentation algorithm is simply an intuitively based approach to the minimization of the error-of-fit function, while recent optimization-based approaches use a more direct minimization strategy.

A statement at the computational level of curve segmentation in terms of fitting allows the task to be restated as an optimization problem as follows. Given:

- A set of 2D *data points* $P = \{\mathbf{x}_i\}_{i=1}^{n}$, where $\mathbf{x}_i = (x_i, y_i)$;

- A *family of curves* $C(\mathbf{a})$ (which may be piecewise collections of simpler curves), parameterized by the vector $\mathbf{a}$;

- An *error metric* $\epsilon^2(C, \mathbf{a}, P)$, which measures the likelihood that the curve $C(\mathbf{a})$ explains the data points P;

find the value $\mathbf{a}_{\min}$ for which the error function attains its global minimum. The curve $C(\mathbf{a}_{\min})$ is then that which *best fits* the data.

This statement of the problem includes the traditionally understood definition of curve segmentation, because the family of curves may be a collection of simpler curves, with the "parameter vector" $\mathbf{a}$ being the concatenation of the parameters of the simpler curves. For example, if we consider Lowe's seminal line segmentation algorithm [72], the curve family chosen is the family of "polylines which interpolate a subset of the data points", parameterized by the point indices of the subset; the error metric for each segment is the ratio of maximum point deviation to the segment length; and the optimization strategy is the familiar recursive maximum-deviation split and merge. Expressing the problem in this form allows us to separate the representation and error metrics from the precise algorithm used to extract the segmentation. For example, we notice that the curve representation used by Lowe must interpolate some of the data points, although there may be a more significant interpretation of the data if this requirement is relaxed [29]. Also, the "greedy" split-and-merge algorithm need not necessarily converge to the global optimum.

1.2.3 Summary: Defining stability

Referring back to the Marr-Brady criteria, I redefine **stability** as the requirement that on presenting the same object to the system at two different instances, from different viewpoints and under different illumination conditions, we should recover the same segmentation. Stability is therefore a combined property of the choice of curve representation and of the algorithm used to extract that representation from the image.

Key to stability is the property that images which differ by small amounts should produce segmentations which differ by small amounts. An unstable technique may produce widely different segmentation results even when given input data that are qualitatively identical. Instability may be a result of either fundamental theoretical problems or just algorithmic issues. For example, in the popular region-growing approach to segmentation, the choice of seed-point may be viewed as an algorithmic issue impacting instability [69], while Figure 1.1 illustrates how the fundamental issue of choice of curve representation incurs similar problems.

1.3 Thesis: The interaction between stability and scope

The central theme of this thesis is the search for stable solutions to the 2D segmentation problem, and the observation that the interaction between stability and scope underpins the choice of segmentation strategies.

While Marr and Nishihara explicitly acknowledge the tradeoff between stability and sensitivity, they do not emphasize their interaction with the requirements of scope and uniqueness. A goal of this thesis is to demonstrate that there is a tradeoff between scope and *stability* in the context of real systems, and that the extent to which this tradeoff is overcome provides a qualitative metric for representations.

The thesis begins with a review of the research into segmentation representations and error metrics, casting the segmentation process as a fitting problem. Following Brady's first criterion, I explore only invertible or fully information-preserving representations, and because of the importance of planar curves in describing the common case of describing a two-dimensional edge image, deal largely with curve representations. In some cases, however, these are extended to surfaces extracted from $2\frac{1}{2}$D range images.

One of the simplest fitting problems, that of fitting ellipses and general conic sections to 2D data, is then explored in detail in Chapters 3 and 4, and a new ellipse-specific fitting algorithm using only linear computations is introduced. The new fitter exhibits significantly greater stability than the more general alternatives to which it is compared, providing an interesting demonstration of how narrowing the scope of a representation increases stability.

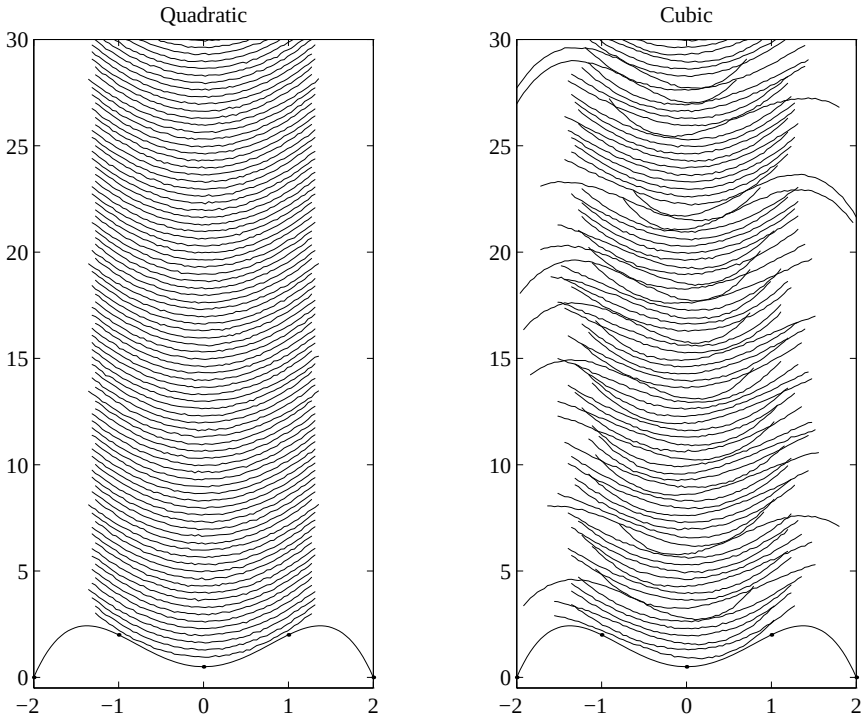


Figure 1.1: Demonstration of instability problem when region-growing high-order curves. Quadratic (left) and cubic (right) polynomials have been fitted to the quartic data at the bottom of each image. For each of 30 runs, the data were corrupted with 2% Gaussian noise and a region growing algorithm was seeded in all instances from the point $(0, \frac{1}{2})$. (The results are depicted by stacking the noise-corrupted datasets vertically and showing only the subset of the data to which the curve was grown). The quadratic curves grow to approximately the same points on each run, while the cubics vary widely. The variation in the cubics' support is due to the equal likelihood that one side or other of the data can provide slightly smaller residuals.

Further specializing to circle-specific fitting emphasises the link.

The second aspect of the scope-stability tradeoff is in the scope of the noise model that a system assumes. This is related to the concept of "algorithmic" stability introduced above, in that variations in the parameters of the initial noise estimate ought not to have a detrimental effect on the output of the system.

Chapter 5 introduces the run-distribution test, a new error-of-fit metric which requires no knowledge of the image noise level and is applicable to a very wide range of noise distributions. Although the scope of the noise model is significantly widened, the sensitivity of the test is reduced. In fact, the standard chi-squared test outperforms the run-distribution test only when its initial parameters are set to within 20% of their correct values. In this case we exchange a great increase in scope for a small reduction in stability.

The exploration of noise-independent error metrics is continued in Chapter 6, this time for piecewise-smooth curves. A critique of the current minimum-description-length (MDL) algorithms allows the development of a segmentation process that locally adapts to changing image noise, again yielding a stable segmentation procedure that widens the range of noise distributions to which it is applicable.

I conclude that addressing the scope-stability tradeoff is an important paradigm for computer vision research, and argue that the techniques described herein demonstrate the utility or validity of this paradigm in a number of different ways.

Chapter 2

Literature Review

This chapter describes the current state of the art in curve segmentation, and in particular the areas of representation, fitting, and extraction. As Chapter 1 has introduced the segmentation problem in terms of the choice of error metric and shape representation, this review centres on these two issues. The chapter begins by introducing some mathematical preliminaries, introducing the problem of curve fitting subject to a distance or error of fit metric. After discussing several such metrics, I describe the principal curve representations used in the vision literature. Smooth curve representations are described first, followed by the piecewise-smooth representations upon which most systems are based.

2.1 Mathematical Preliminaries

Before discussing the details of curve representation in vision, this section provides a brief introduction to the notation and terminology used throughout the discussion.

2.1.1 Parametric and implicit curves

In the mathematical treatment of curves and surfaces it is often convenient to divide the representation into the two classes of implicit and parametric curves. Briefly, a parametric curve is represented in the form

$$C(\mathbf{a}) = \{\mathbf{c}(\mathbf{a};t) \mid t_{\min} \leq t \leq t_{\max}\}$$

where $\mathbf{c}(\mathbf{a};t)$ is the vector function

$$\mathbf{c}(\mathbf{a};t) = \begin{pmatrix} c_x(\mathbf{a};t) \\ c_y(\mathbf{a};t) \end{pmatrix}$$

Parametric curves have the advantage that they are relatively easy to generate, and are thus the representation of choice for computer graphics and computer-aided design applications.

Sometimes, however, a more powerful representation may be provided by considering *implicit* curves, defined as the zero set of a scalar function $F(\mathbf{x})$:

$$C(\mathbf{a}) = \{\mathbf{x} \mid F(\mathbf{a};\mathbf{x}) = 0\}$$

Such curves include the conic sections and superellipses. Although elimination theory [118] may be used to convert between parametric and implicit representations of the same curve, it is common that the complexity of the parametric representation will exceed that of the implicit. Conic sections, for example, are represented naturally by a biquadratic implicit polynomial, but their parametric forms are either transcendental or rational polynomial [51].

2.1.2 Least squares fitting of smooth functions

By far the commonest method of approximating image and edge data by parametrized functions[1] is the technique of least-squares fitting. In the formulation of the introduction, the family of curves from which the approximation is to be drawn is represented by $C(\mathbf{a})$. In this notation, C is used to refer to the entire family, while a particular curve $C(\mathbf{a})$ or $C_{\mathbf{a}}$ is identified with the infinite set of points comprising the realization of the curve defined by the parameter vector $\mathbf{a}$. We may generally define a distance function

$$D(\mathbf{x}; C(\mathbf{a}))$$

which measures how far a point $\mathbf{x}$ is from the curve $C_{\mathbf{a}}$. Where the family of curves is obvious from the context, we write this simply as $D(\mathbf{x}; \mathbf{a})$. We note also that this distance function may not necessarily be based on the minimum geometric distance from the point to the curve. The least-squares problem then is to minimize the least-squares error metric

$$\epsilon^2_{\mathrm{LS}}(C, \mathbf{a}, P) = \sum_{i=1}^{n} D(\mathbf{x}_i; \mathbf{a})^2$$

where P is the set of n data points $\{\mathbf{x}_i\}_{i=1}^n$ as in Chapter 1. The various choices of distance metric and their implications are what distinguish least-squares approaches, and the particular family of curves under consideration will in turn influence such choices.

2.2 Error-of-fit metrics

In the computational theory of Chapter 1, the choice of error-of-fit metric (EFM) is one of the two defining characteristics of a segmentation process. In this section, I discuss the EFMs most commonly used in the vision literature. Such metrics are generally divided into two classes, depending on whether they are used to guide the segmentation directly or simply as a verification stage afterwards. This review concentrates on the former type, as the latter will be covered in Chapter 5. The metrics covered are:

- The *geometric distance*, which is the maximum-likelihood estimator under some noise assumptions.
- Sampson's *gradient distance*, a first order approximation to the geometric distance, the usage of which lightens the computational burden significantly.
- *Algebraic distance*, an approximation for implicit functions which can lead to great reductions in the computational complexity of fitters, in some cases yielding closed-form solutions to the fitting problem.
- Kanatani's *"statistical" distance*, on the other hand, argues that the geometric distance is not the maximum-likelihood estimator in cases where the curve to be fitted is digitally sampled.

Finally the augmentation of these metrics to deal with outliers, yielding statistically *robust* fitting algorithms, is discussed.

2.2.1 Geometric distance

Conceptually, the simplest distance metric between a point and a curve is the distance between the point and its closest point on the curve:

$$D_G(\mathbf{x}, C) = \min_{\mathbf{p} \in C} \|\mathbf{p} - \mathbf{x}\|$$

[1] In this case, the term *parametrized* is used to signify that the function is defined by a (generally small) number of scalar parameters

This metric, the *geometric* distance, is frequently cited by authors
[80, 115, 12]
[113] as the optimal measure to minimize from a least squares point of view, but is generally computationally expensive to compute. In this case, optimality is generally assumed to mean that the curve which minimizes mean geometric distance maximizes the *a-posteriori* probability that that curve is a sampled observation of an ideal curve corrupted by Gaussian noise. Such an estimate is known as a maximum *a-posteriori* (MAP) or maximum-likelihood (ML) estimate. However the optimality of this estimate depends [60, §29] on the assumption that each observed point is the image of only one curve point, which as discussed in §2.2.4 may not be a reasonable model.

2.2.2 Algebraic distance

Implicit curves introduce a natural distance metric from any point on the plane to the curve. The *algebraic distance* from the point $\mathbf{x}$ to the curve defined by $F(\mathbf{x}) = 0$ is simply the value of F at $\mathbf{x}$:

$$D_A(\mathbf{x}, C) = F(x)$$

We note that the algebraic distance will not in general exhibit a simple relationship to the geometric distance, and indeed may be dependent on the scaling of the parameterization of C. The principal advantage of this distance metric is that it greatly simplifies the computation in many algorithms and thus it is widely used.

2.2.3 Gradient distance

The gradient distance (often referred to as Sampson's distance after [115]) is defined for implicit curves $F(\mathbf{x}) = 0$ as

$$D_\nabla(\mathbf{x}, C) = \frac{F(\mathbf{x})}{\|\nabla F(\mathbf{x})\|}$$

It is derived as a first-order approximation to the geometric distance and while generally much simpler to compute, remains moderately expensive to use directly in approximation problems.

2.2.4 Kanatani's "Statistical distance"

Recently, though, Kanatani [55] has argued that geometric distance is not in fact the MAP estimator for curves other than the line. His argument is illustrated in Figure 2.1, and a variation of it will be presented here. If we can model the imaging and edge extraction process by assuming that a true curve is digitally sampled, and that a random error vector taken from an isotropic Gaussian distribution is then added to each sampled point, then the probability distribution function (pdf) of the imaged curve points is a Gaussian blur of the original curve. This is equivalent to saying that the value of the pdf at any point is the sum of the contributions from all the individual independent Gaussians. In Bayesian terms, the likelihood of observation of a particular point $\mathbf{p}$ is the probability that any point on the curve might be perturbed to give rise to that point. Applying the identity

$$P(A \text{ or } B \mid C) = P(A \mid C) + P(B \mid C) - P(A \text{ and } B \mid C)$$

and using the assumption of independence of the individual error vectors to set $P(A \text{ and } B \mid C) = 0$ yields the probability of observing point $\mathbf{p}$, given the curve C and the noise pdf[2] G

$$P(\mathbf{p} \mid C) = \oint_{\mathbf{x} \in C} G(\mathbf{x} - \mathbf{p})$$

[2] Here we assume that the Gaussian function G has been scaled to ensure that the resultant probabilities sum to 1.

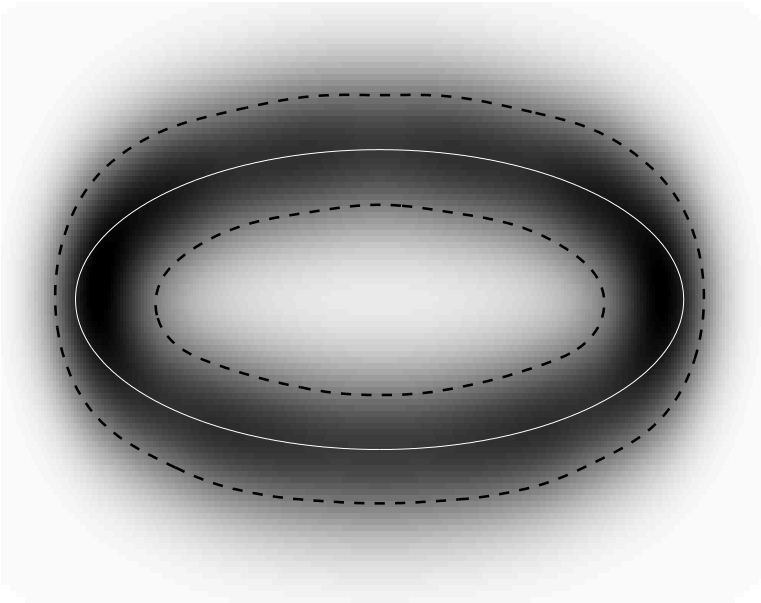


Figure 2.1: Kanatani's case against geometric distance. The picture shows a digital ellipse (white line) convolved with a Gaussian filter. The intensity value at every point is then a discrete approximation of the probability that that point might have been sampled from the ellipse. (Darker points are more probable.) Black dashed lines mark the boundaries of the 68% confidence region. Note that on the high curvature sections, these boundaries are much closer to the curve on the outside than on the inside. On these sections, points with equal geometric distances from the ellipse are *not* equally probable (or equivalently, points with equal probability of observation are not equidistant from the ellipse) and hence minimization of mean geometric distance is not the maximum-likelihood estimator.

Given an implicit curve $F(\mathbf{x}) = 0$, evaluation of this function at point $\mathbf{p}$ is equivalent to the convolution $\iint \delta(F(\mathbf{x}))G(\mathbf{x} - \mathbf{p})\mathrm{d}A$, where δ is the Kronecker delta. Further specializing to the case of discrete data, this allows us to calculate the value of the probability at every pixel in an image by convolving a binary image of the curve with a Gaussian kernel of appropriate σ.

Figure 2.1 illustrates the value of this probability distribution for a particular sampled ellipse. We note that in the high curvature areas the points *inside* the ellipse receive larger contributions from the surrounding points than do those outside. Given two points equally distant from the true ellipse, then, the point inside is more likely to have resulted from our imaging model than the outer point. Thus, minimizing the sum of geometric distances gives an excessive weighting to points outside the ellipse. In addition, the weighting is different at the high and low-curvature regions, giving uneven weighting to points in these regions.

Of course, this argument immediately begs the difficult question "Is this noise model appropriate for the final output of the complex processes of image acquisition and edge extraction". This is a difficult problem. Although the original work by Canny [22, 21] includes an analytic model of the localisation error inherent in his edge detector, implementations of the Canny operator use an approximation to the ideal operator. Further, the addition of non-maximal suppression and subpixel interpolation stages further complicate the analysis of a real algorithm.

I have taken a more empirical approach, noting simply that more noisy images require higher values of the Canny σ parameter. This is a true statement regardless of the image

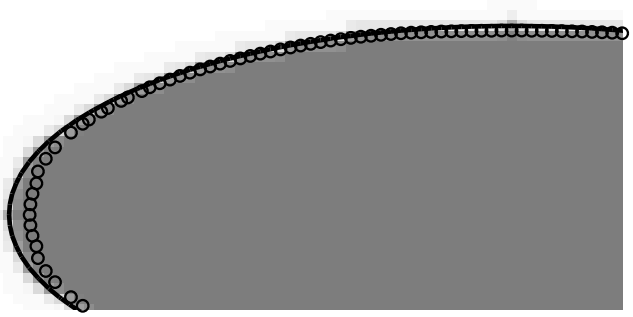


Figure 2.2: Verification of the Gaussian-distributed residuals assumption in Figure 2.1. The synthetic ellipse image is processed by a Canny edge detector with σ = 5 pixels. Subpixel edgel sites reported after non-maxima suppression are plotted as circles. We note again the bias toward the center of curvature, as modelled by the isotropic Gaussian assumption; supporting both the noise model and the argument for "statistical" rather than geometric distance.

noise as it simply describes the approach taken by researchers when extracting edges from a noisy image. In that case, we may observe the effect of large σ on a synthetic image with known ground truth and verify qualitatively whether Kanatani's conclusion is correct. Figure 2.2 shows a synthetic ellipse image, generated from the implicit equation of an ellipse $F(\mathbf{x}) = 0$ by blurring the binary image given by $I_{ij} = F(i,j) > 0$. On the image are plotted the edgel sites reported by a subpixel Canny operator after non-maxima suppression. We note that the locus of edgel points is similar to that predicted by the sum-of-Gaussians noise model. Finally, Figure 2.3 demonstrates the phenomenon on a real image.

While this is not yet unequivocal support for the noise model, it allows us to infer that MAP estimation over this probability distribution will provide a better model of point distributions arising from Canny edgel extraction on real images than that implied by the geometric distance. *It also allows us to generate synthetic ellipse data simply by sampling an analytic ellipse and adding isotropic Gaussian noise.*

Kanatani describes the distance metric induced by this model as "statistical distance", and presents a bias correction algorithm which approximates the minimization of this distance, which will be compared with other estimators in Chapter 4.

2.2.5 Robust metrics

All the previous measures may be viewed as approximations to the maximum-likelihood estimators given an underlying noise model which is Gaussian or very similar to Gaussian. While these models are often derided, they are quite adequate for the modelling of the small fluctuations in feature positions due to inaccuracies in the imaging process. What they fail to model well are the gross fluctuations caused by phenomena outside the usual imaging pipeline. The result of fitting data which is corrupted with such *outliers* is generally far from that which is desired. Robust fitting [53, 76, 128] modifies the preceding distance metrics to reduce the influence of points which fall too far outside the fitted curve. As robust metrics feature in Chapter 6, a brief summary is included here for completeness.

Huber's treatment of robust fitting collapses the general problem down to that of finding

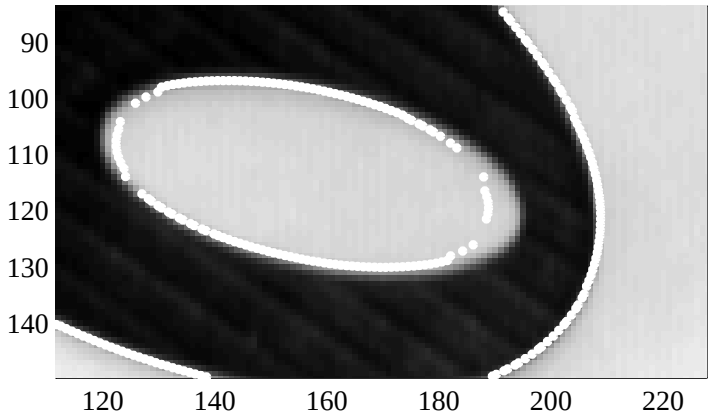


Figure 2.3: Canny edgels on a real image, again illustrating the bias toward the center of curvature caused by smoothing with a large σ.

the *mode* of a one-dimensional distribution. In the language of vision, this is equivalent to fitting a curve of constant height $y(a;x) = a$ to a 1-d set of sampled data $\{y_i\}$. The standard non-robust estimator of the mode a is simply the mean of the data

$$a = \frac{1}{n}\sum y_i$$

In order to generalize this to curve and surface fitting in higher dimensions, we note that the mean is the value a_m which minimizes the sum of squared residuals $\sum(y_i - a_m)^2$. Finding the "mean" of the data then corresponds to the least squares problem as defined above: finding the set of parameters $\mathbf{a}$ that minimize the sum of squared distances between the sampled points and the curve defined by $\mathbf{a}$. Returning to the task of estimating the mode, the median is a more robust estimator, minimizing the sum of *absolute deviations* $\sum|y_i - a|$.

As a proof of this, based on [93, page 703], note that the derivative of $|x|$ is plus or minus 1 depending whether x is positive or negative. The minimum of $\sum|y_i - a|$ occurs when its derivative is zero, or when the sum of signs $\sum \mathrm{sgn}(y_i - a)$ is zero. This sum simply counts the difference between the number of y_i greater than a and the number less than a, so it is zero when exactly half of the y_i are above a and half below, which is the definition of the median.

A more direct approach is to directly search for the mode by binning the y_i into a histogram and looking for the most populated bin. This is the technique well known as the *Hough transform* [52], applied to the problem of one-dimensional fitting. Its equivalence to an optimization problem [62] is obtained by noting that finding the bin with the greatest number of votes is equivalent to minimizing the sum $\sum\phi_a(y_i)$ where ϕ_a is the inverse "top-hat" function

$$\phi_a(y) = \begin{cases} 0 & \lfloor a/w \rfloor < y < \lfloor a/w \rfloor + 1 \\ 1 & \text{otherwise} \end{cases}$$

where w is the width of the bins used in the equivalent histogram.

These robust techniques are particular examples of *M-estimates*, where the least squares *kernel* $x \to x^2$ is replaced by an robust kernel or *influence function* $x \to \phi(x)$. Figure 2.4 illustrates a few sample influence functions, showing how they tend to place less weight on data far from the mean than the least squares kernel.

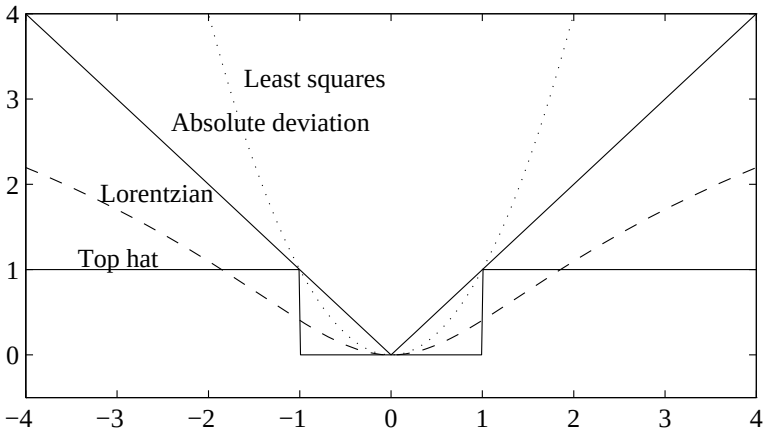


Figure 2.4: Robust kernels. The definitions of the functions are in Table 2.1. For clarity the Huber function is not shown as it would obscure the top hat and least squares graphs.

Function	Definition	Estimator
Least-squares	x^2	Mean
Absolute deviation	$\|x\|$	Median
Top hat	$\|x\| > 1$	Binned mode
Lorentzian	$\log(1 + \frac{x^2}{2})$	–
Huber	$\begin{cases} 1 & \text{if } \|x\| > 1 \\ x^2 & \text{otherwise} \end{cases}$	–

Table 2.1: Robust kernels. Graphs of the functions are shown in Figure 2.4 and the corresponding estimator of central tendency is added where it has a simple solution.

Representation	Scope	Stability	Accessibility
Polyline	Poor	Fair	Good
Conic	Fair	Good	Good
Polyconic	Good	Fair	Fair
Superellipse	Fair	Fair	Poor
Parametric Polynomial	Good	Poor	Good
Implicit Polynomial	Good	Poor	Fair
B-Splines	Excellent	Poor	Fair

Table 2.2: Evaluation of several curve representations using the Marr-Brady criteria. The "grades" awarded reflect the extent to which each representation favourably meets the particular criteria. The line, conic and superconic are all intrinsically scope-limited representations, while the others offer a variety of tradeoffs, notably against stability, in order to achieve wide scope.

To conclude this section, I note that the concepts of robustness and the choice of EFM are largely orthogonal. A particular choice of EFM represents the *a priori* assumptions about the properties of the statistical distribution from which the inliers are drawn, and can generally be augmented by the robust kernel of choice to indicate the *a priori* assumptions about outliers.

2.3 Criteria for comparing shape representations

A variety of shape representations have been proposed for computer vision, some borrowed from computer graphics, symbolic AI and approximation theory; some specifically directed toward vision. In turn, we note two main varieties of representation: geometric representations such as parametric and piecewise linear curves; and property-based representations such as the predicate descriptions of [54, 121], the schema in VISIONS [47], or sparse geometric information used for geometric hashing [64, 122, 112]. Between these varieties lie a number of more exotic representations such as Fourier descriptors [5] and codons [49].

As this thesis concentrates on geometric curve representations rather than the property based approaches, I initially review several common curve types used in vision and compare them on the basis of the Marr-Brady criteria. Table 2.2 summarises these results in the form of a report card. The "grades" given are elaborated in the following sections. I will begin however, by critiquing and clarifying the Marr-Brady criteria as they apply in this work.

The previous chapter has already introduced the Marr-Brady criteria in order to present the motivation for this thesis. However recent research in the field, in conjunction with advances in computing hardware, has meant that these criteria demonstrate certain inadequacies. These will be discussed as the following paragraphs introduce the "grading" system for Table 2.2. The original Marr-Brady criteria presented above are subsumed into just three categories: scope, stability and accessibility; and the representations are evaluated according to these three criteria.

Scope

Marr's initial definition of scope is careful to mention the domain of applicability of the modelling system. In the case of a general vision system, we simply require that the scope is "large". A representation with excellent scope allows the representation of arbitrary shapes to within a certain degree of accuracy. Lower grades are intrinsically limited, but may still have wide scope.

Stability

The stability criterion measures how variations in the input data such as noise and occlusion affect the canonical description or uniqueness of the final fitted representation. Marr's definition of uniqueness is rather stronger than that currently viewed as necessary in vision. It fails to note that the representation may simply be *covariant* with transformations provided that the canonical descriptions can be readily extracted. As an example, he notes that n^{th} degree polynomial models have a different formal description depending on the coordinate system chosen. However, the work of [61, 124] on polynomial invariants illustrates that the uniqueness criterion may be relaxed to that of requiring that invariants be readily extracted from the representation.

Both Marr's stability criteria and Brady's local support may be seen as dealing with the effects of noise and occlusion. This discussion combines the two effects under one criterion, measuring how both the intrinsic and extrinsic invariants of the representation are affected by errors in the input data. Although the relative effects of noise and occlusion on different representations may be markedly varied, this combined criterion is nevertheless a reasonable measure of the usefulness of a representation for later operations in a vision system.

Accessibility

Marr defines accessibility to mean that the representation can be extracted from the input data in reasonable time. In this discussion, the highest accessibility rating is given to representations which may be extracted by a closed-form or well understood operation such as eigenvector extraction. Techniques requiring iterative minimization or combinatorial optimization are rated lower.

2.4 Curve representation in vision

In this section I first review the representations that have been proposed for curves which have a simple parametric description. I then consider the descriptions which comprise piecewise collections of the simple curves.

2.4.1 Lines

Since the seminal work of [102], line segments have been used as a representational basis for vision. In a blocks world such as that assumed by the Roberts system, lines provide the richest representation necessary and are unique and stable. Within the blocks world, line representation has been well studied and the use of techniques such as the Hough transform [52] well explored. In the curved world of modern vision however, they are inadequate. Although curved edges may be approximated to an arbitrary degree of accuracy by piecewise linear segments, the resulting descriptions are inherently unstable. Despite this, research is still active into the problems of line segmentation [29, 70, 71, 100, 107], indicating that even for this relatively simple case, a "best practice" algorithm has not been agreed.

2.4.2 Conic sections

The importance of conic sections in computer vision has always been recognised, as they arise naturally as the perspective projection of the circle. They are also one of the simplest curve representations. The increasing interest in the projective invariants of conics [96, 56, 94, 95, 23] has recently reaffirmed their importance. As the next chapter gives a more detailed review of conic fitting techniques, here I will just touch on the solutions that have been proposed.

Conics were used as a contour representation in [82, 2]; but the paper generally credited with the introduction of conic sections to vision is [16], in which the algebraic distance is minimized subject to a quadratic constraint on the conic parameters.

Before reviewing the literature on general conic fitting, we will introduce a statement of the problem that allows us to unify several approaches under the umbrella of constrained least squares. Let us represent a general conic by an implicit second order polynomial:

$$F(\mathbf{a}, \mathbf{x}) = ax^2 + bxy + cy^2 + dx + ey + f = 0, \tag{2.1}$$

where $\mathbf{a} = [a\ b\ c\ d\ e\ f]^T$ and $\mathbf{x} = (x\ y)^T$. $F(\mathbf{a}; \mathbf{x}_i)$ is called the "algebraic distance" of a point $\mathbf{x}_i$ to the conic $F(\mathbf{a}; \mathbf{x}) = 0$. The fitting of a general conic may be approached [49] by minimizing the sum of squared algebraic distances

$$\mathcal{D}_A(\mathbf{a}) \ = \ \sum_{i=1}^{N} F(\mathbf{a}; \mathbf{x}_i)^2 \tag{2.2}$$

of the curve to the N data points $\mathbf{x}_i$. In order to avoid the trivial solution $\mathbf{a} = \mathbf{0}_6$, and recognizing that any multiple of a solution $\mathbf{a}$ represents the same conic, the parameter vector $\mathbf{a}$ is constrained in some way. Many of the published algorithms differ only in the form of constraint applied to the parameters:

- Most authors cited below mention the most "obvious" normalization $\|\mathbf{a}\|^2 = 1$ (used in [82, 14]) as a strawman. This constraint has the advantage that all conics may be scaled to satisfy it, but fails to be invariant even to Euclidean transformation of the data.

- Gnanadesikan [42] uses $a^2+b^2+c^2+d^2+e^2 = 1$ which excludes only the non-conic $1 = 0$, but again is non-invariant.

- The normalization $f = 1$ used in [9, 2, 25, 30] was also investigated in [92] and [104]. This normalization does not permit the fitting of ellipses which pass through the origin, and is not invariant. However Rosin points out that given conditioned or rescaled data (c.f. §4.5), this normalization tends to favourably correct for the bias towards high curvature.

- In [104, 91, 40] the constraint used is $a + c = 1$, eliminating from consideration the family of hyperbolae $x^2 - y^2 = k$. This constraint is invariant to similarity transforms.

- Bookstein [16] proposes $a^2 + \frac{1}{2}b^2 + c^2 = 1$, invariant to similarity transforms of the axes and allowing all conics to be represented but excluding lines, for which $A = B = C = 0$.

- Forsyth *et al.* [38] proposes the *cubic* constraint $\det(A) = 1$, where

$$A = \begin{pmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{pmatrix}.$$

  Although they claimed that minimization subject to this constraint results in projectively invariant fitting, Kapur and Mundy [57] demonstrate that the fitting is in fact only affine invariant.

- Taubin's approximate square distance [124] may also be viewed as the quadratic constraint $\|N\mathbf{a}\|^2 = 1$ where N is the Jacobian $[\nabla F(\mathbf{a}; \mathbf{x}_1) \ldots \nabla F(\mathbf{a}; \mathbf{x}_N)]^T$.

Note that these constraints are all either *linear*, of the form $\mathbf{c} \cdot \mathbf{a} = 1$ or *quadratic*, constraining $\mathbf{a}^T\mathbf{C}\mathbf{a} = 1$ where $\mathbf{C}$ is a 6×6 *constraint matrix*.

General conic fitting

The seminal work by Bookstein [16] introduces the invariant constraint $a^2 + \frac{1}{2}b^2 + c^2 = 1$. He showed that this leads to the solution of a rank-deficient generalised eigenvalue problem for which he gives an efficient solution by block decomposition.

Sampson [115] presents an iterative improvement to the Bookstein method which replaces the algebraic distance $F(\mathbf{a};\mathbf{x})$ with a better approximation to the geometric distance, called the *gradient distance*:

$$\mathcal{D}_S(\mathbf{a}) \;=\; \sum_{i=1}^{N} \frac{F(\mathbf{a};\mathbf{x}_i)^2}{\|\nabla_{\mathbf{x}} F(\mathbf{a};\mathbf{x}_i)\|^2} \tag{2.3}$$

In his implementation, Sampson further simplified this metric by recasting its minimization as an iterated reweighted least squares problem: using the value of $\mathbf{a}$ from the previous iteration to calculate the gradient at the current iteration. Sampson shows how the use of this new distance measure increases the stability of the fitting, but because it necessitates an iterative algorithm, the computational requirements are substantially increased.

Taubin [124] proposed a further approximation of (2.3) as

$$\mathcal{D}_T(\mathbf{a}) \;=\; \frac{\sum_{i=1}^{N} F(\mathbf{a};\mathbf{x}_i)^2}{\sum_{i=1}^{N} \|\nabla_{\mathbf{x}} F(\mathbf{a};\mathbf{x}_i)\|^2}, \tag{2.4}$$

which, while strictly valid only for a circle, again allows the problem to be expressed as a generalized eigensystem, reducing the computational requirements back to the order of Bookstein's process.

2.4.3 Superellipses

Although conic sections are useful in many industrial applications as a curve representation, they lack scope when applied to natural objects and scenes. Introduced to the computer graphics community in [6] (who refers in turn to Gardiner's column in *Scientific American* [41]), superellipses are a natural generalization of the ellipse $ax^2 + by^2 = 1$ allowing exponents different from 2:

$$f(x,y) = \left(\frac{x}{R_x}\right)^{\frac{2}{\epsilon}} + \left(\frac{y}{R_y}\right)^{\frac{2}{\epsilon}} = 1$$

Superellipses became useful in vision when algorithms for the fitting of superellipses and their 3D surface equivalent, the superquadric, were introduced by [86]. Pentland's original algorithm minimizes the squared algebraic distance

$$\epsilon^2 = \sum_{i=1}^{N} (f(x,y) - 1)^2$$

As in the case of simple conic fitting, however, this metric proves sensitive to occlusion. In order to overcome this [120] proposed an area-minimizing formulation:

$$\epsilon^2 = \sum_{i=1}^{N} R_x R_y (f(x,y) - 1)^2$$

By thus biasing the minimizing superellipses to have low area, performance in the presence of occlusion is improved. Segmentation of range images into superquadric patches has been well studied [68, 46, 97], however few authors [109, 89] have used superellipses as a curve representation. Superellipses extend the scope of the conic sections, particularly when the basic representation is augmented with Pentland's deformations of bending and

tapering. In addition, this increase in scope is achieved without a significant reduction in local extrapolation abilities. Stability is, however, adversely affected due to the strongly nonlinear form of the minimization. Solina's modification, while ameliorating this effect, does not eliminate it entirely. Finally, the cost of superellipse fitting is significantly greater than that of conic fitting due to the requirement for iterative optimization procedures. A recent development [90], the application of point-distribution models to superellipses, may however reduce this cost considerably.

2.4.4 Parametric polynomials

The most important applications of parametric polynomial curves and surfaces in vision are the modelling of active contours using B-splines [58], and the use of biquadratic polynomials for range image segmentation [8, 27, 69]. The fitting of the one-dimensional polynomial of order p

$$y = a_0 + a_1 x + a_2 x^2 + \cdots + a_p x^p$$

is readily achieved [93] if we assume that the errors in observed points are only in the y coordinates. This may be the case, for example, when sampling a one dimensional signal at known intervals. Two dimensional curves admitting a parametric form $\mathbf{c}(t)$, which have been sampled to give discrete points $(x_i, y_i) = \mathbf{c}(t_i)$ may be fitted by treating each component separately:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_{x0} + a_{x1} t + a_{x2} t^2 + \cdots + a_{xp} t^p \\ a_{y0} + a_{y1} t + a_{y2} t^2 + \cdots + a_{yp} t^p \end{pmatrix}$$

It can be shown [118] that when the order of the polynomials is limited to quadratic, this procedure fits a parabola of arbitrary orientation to the data, independent of the choice of t_i. Given a list of points in order around the curve, the sample values may then simply be set to the point indices: $t_i = i$. While such polynomials can approximate any curve to arbitrary accuracy by increasing the order of the polynomial, they become increasingly unstable [35].

2.4.5 Implicit polynomials

Implicit polynomials, also known as algebraic curves and surfaces, are a shape representation that have been recently introduced to the vision community [123, 24, 124, 12, 38, 61, 126]. Algebraic curves are implicit curves of the form

$$F(x, y) = 0$$

where the function F is polynomial in x and y. Thus, the general conic is a degree 2 algebraic curve. Taubin's original paper suggested the use of high-order algebraic curves as an alternative to piecewise collections of simpler curves, and introduced the *approximate mean square* distance measure

$$\frac{\sum F^2(x, y)}{\sum \|\nabla F(x, y)\|^2}$$

as a computationally feasible alternative to the gradient-weighted distance. This provides an attractive alternative to piecewise representations: one implicit polynomial represents all segments of the curve simultaneously, and may be fit to the data efficiently without requiring the search for knot points.

Although the precise coefficients of the algebraic polynomial will vary with the orientation and scaling of the input data, it is possible to extract invariants from the representation [39] which may be used for matching, thus satisfying the uniqueness constraint. However, research into the noise sensitivity of these approaches [129] suggests that to overcome noise sensitivity, a considerable amount of work is required.

It is also interesting to note that implicit curves and surfaces have difficulty in satisfying Brady's invertibility requirement, for two reasons. First, invertibility is effectively equivalent to being able to draw the curve given its representation in terms of the coefficients; however, the drawing of such curves is still an active area of computer graphics research [51, 125], implying that recovering the original data from the representation is difficult. The second, smaller, difficulty arises because the canonical representation of high-order algebraic curves includes no endpoint information, resulting in a loss of information about the finite extent of the curves. However, this information can be trivially included as an adjunct to the pure coefficients.

2.5 Piecewise representations

In general, complex curves are represented not by a single analytic curve, but *polycurves*: the piecewise join of several segments, each of which is described analytically. If continuity conditions are imposed on the breakpoints (or *knots*) of the polycurve, the curves are known as splines. Such a curve may be described by its set of knot points and the set of parameter vectors associated with each segment. As such a representation is generally overspecified, it is common in the case of splines to represent each segment by a number of control points sufficient to uniquely determine the shape of the curve, thus providing a minimal parameterization.

All polycurves expand considerably the scope of representation of their constituent curves, but unless the knot positions are clearly defined by tangent or other discontinuities, the stability of the representation will be reduced — a polyline may approximate a circle to any accuracy, but the position of the breakpoints may be offset by an arbitrary angular amount and still achieve the same error.

With polycurves, the equivalence of the segmentation problem and the fitting problem becomes more readily apparent—fitting a polycurve involves the estimation both of the positions of the knot points and of the parameters of the approximating segments. This is known as the optimal knot placement problem in spline fitting [28] and no efficient solution is currently known [117].

B-Splines

Cubic B-splines are among the most popular curve representations in computer graphics, being capable of modelling a large number of the curves occurring in computer-aided design. As such, they satisfy the scope requirement for a large class of problems. In computer vision, the fitting of splines is often cast in the energy-minimizing snake formulation [127], and splines have proved useful for tracking and curvature estimation. Although the extraction of invariants from snake-approximated boundaries and the introduction of the affine snake [10] begin to address the invariance requirement, there is still little work done on object recognition using such measures ([116] is an example of the state of the art).

Conic splines

Conic splines, introduced by Bookstein [16] and developed by Pavlidis [83, 84] are collections of conic sections with first and second order continuity at the joins. Bookstein provides an efficient method to fit such splines to point data, given the positions of the knot points, but does not address the problem of knot placement. Pavlidis introduces a control-point representation which allows the local shape of the spline to be altered.

2.6 Summary

This section has reviewed the state of current research into curve representation and fitting in computer vision. I have described the principal curve representations used and compared them under the headings of scope, invariance and accessibility. These often competing requirements have yet to admit a universal solution, with different applications imposing different choices. It does seem reasonable to say that most higher-level vision systems have to date limited the range of curves with which they deal to piecewise collections of conics and straight lines.

I have described the approaches taken to the fitting of curves, both singly and multiply parameterized. These approaches may be broadly classified by the error-of-fit measure that they use and more precisely by the techniques used to minimize these measures.

Chapter 3

Least-Squares Fitting of Ellipses

This chapter introduces the problem of least-squares ellipse fitting. First we review the existing work on the subject, and then present a new efficient method for fitting ellipses to scattered data. Previous algorithms either fitted general conics or were computationally expensive. By minimizing the algebraic distance subject to the constraint $4ac - b^2 = 1$ the new method incorporates the ellipticity constraint into the normalization factor, transforming a difficult inequality-constrained problem to a simple equality-constrained one.

A major part of this chapter is spent in a thorough experimental and theoretical examination of several conic-fitting algorithms, including the ellipse-specific one. Finally, simple extensions to the problem of recovering hyperbolae are introduced.

This chapter describes work done in collaboration with Maurizio Pilu. The ellipse-specific constraint was introduced by me in [36], but merely presented as a curiosity. Pilu noted the stability of the method, and developed the theoretical proof of the algorithm's performance. The proof presented here corrects a difficulty with that in [88], but depends crucially on his key lemma, which is presented in Appendix A.

3.1 Introduction

One of the most commonly used curves in vision is the ellipse which, being the perspective projection of the circle, is of great importance for many industrial applications. Despite its importance, however, there has been until now no computationally efficient ellipse-specific fitting algorithm [104, 40].

This chapter introduces a new method of fitting ellipses, rather than general conics, to segmented data. As we shall see in the next section, current methods are either (a) computationally expensive: requiring the equivalent of many (5 to 100) iterations of one of the linear algorithms; or (b) perform ellipse fitting by least-squares fitting to a general conic and rejecting non-elliptical fits. In fact, both [134] and [109] state that ellipse-specific fitting is excessively computationally expensive.

These latter methods are cheap and perform well if the data belong to a precisely elliptical arc with little occlusion but suffer from the major shortcoming that under less ideal conditions — non-strictly elliptical data, moderate occlusion or noise — they often yield unbounded fits to hyperbolae. In a situation where ellipses are specifically desired, such fits must be rejected as useless.

A number of iterative refinement procedures [115, 55, 91] alleviate this problem, but do not eliminate it. In addition, these techniques often increase the computational burden unacceptably.

The new fitting method combines the following advantages:

- *Ellipse-specificity*, providing useful results under all noise and occlusion conditions.

- *Invariance* to Euclidean transformation of the data.
- High robustness to noise.
- High computational efficiency.

Section 3.5 and the following chapter contains experimental results, notably to highlight noise resilience, invariance properties and behaviour for non-elliptical data. I conclude this chapter by presenting some possible extensions.

3.2 Previous Methods and their Limitations

The literature on ellipse fitting divides into two general techniques: clustering and least-squares fitting.

Clustering methods are based on mapping sets of points to the parameter space, such as the Hough transform [65, 133] and RANSAC [14] or accumulation methods [103]. These Hough-like techniques have some great advantages, notably high robustness to occlusion and no requirement for pre-segmentation, but they suffer from the great shortcomings of high computational complexity and non-uniqueness of solutions, which can render them unsuitable for real applications. Particularly when curves have been pre-segmented, their computational cost is significant.

Least-squares techniques center on finding the set of parameters that minimize some distance measure between the data points and the ellipse. In this section I briefly present the most cited works in ellipse fitting and its closely related problem, conic fitting. It will be shown that the direct specific least-square fitting of ellipses has, up to now, not been solved.

3.2.1 Towards ellipse-specific fitting

A number of papers have concerned themselves with the specific problem of recovering ellipses rather than general conics. Bookstein's method does not restrict the fitting to be an ellipse, in the sense that given arbitrary data the algorithm can return an hyperbola or a parabola, even from elliptical input, but it has been widely used in the past decade.

Porrill [91] and Ellis *et al.* [30] use Bookstein's method to initialize a Kalman filter. The Kalman filter iteratively minimizes the gradient distance (2.3) in order to gather new image evidence and to reject non-ellipse fits by testing the discriminant $b^2 - 4ac < 0$ at each iteration. Porrill also gives nice examples of the confidence envelopes of the fittings.

Rosin [104] also uses a Kalman Filter, which is constrained to be ellipse specific by initializing the filter with a circle. If the filter converges to a hyperbola when given a particular data set, the initial covariance of the state vector is increased relative to that of the data points. Optimizing the ratio of initial to data covariance yields the elliptical fit that is least biased towards the circular initial state.

He also [104] analyses the pro and cons of two commonly used normalizations, $f = 1$ and $a+c = 1$ and shows that the former biases the fitting to have smaller eccentricity, therefore increasing the probability of returning an ellipse, at the cost of losing transformational invariance.

Although these methods transform the disadvantage of having a non-specific ellipse fitting method into an asset by using the ellipse constraint to check whether new data has to be included or to assess the quality of the fit, the methods require many iterations in the presence of very bad data, and may fail to converge in extreme cases.

Recently Gander *et al.* [40] published a paper entitled "Least-square fitting of ellipses and circles" in which the normalization $a + c = 1$ leads to an over-constrained system of N linear equations. The proposed normalization is the same as that in [91, 108] and it does not force the fitting to be an ellipse (the hyperbola $3x^2 - 2y^2 = 0$ satisfies the constraint).

Of course, any minimization-based approach that explicitly parameterizes the ellipse—using center, orientation and radii for example—is inherently ellipse-specific, and minimization of the geometric distance [80] is one such approach. Gander[40] provides an efficient technique for performing this minimization. Rather than optimizing over the five ellipse parameters and then solving for the geometric distance at each data point, Gander *et al.* incorporate an extra θ_i parameter for each data point, which defines the closest point on the ellipse. The error function is then much simpler than in the 5-parameter case, at the cost of minimizing over a $5+n$ parameter space. Despite this innovation however, minimization of geometric distance remains one of the most computationally expensive approaches.

Haralick [49, §11.10.7] takes a different approach. Effectively, he guarantees that the conic is an ellipse by replacing the coefficients $\{a, b, c\}$ with new expressions $\{p^2, 2pq, q^2 + r^2\}$ so that the discriminant $b^2 - 4ac$ becomes $-4p^2r^2$ which is guaranteed negative. Minimization over the space $\{p, q, r, d, e, f\}$ then yields an ellipse. His algorithm is again iterative, and an initial estimate is provided by a method of moments. Keren *et al.* [61] apply a similar technique to Haralick's and extend the method to the fitting of bounded quartic curves. Again, their algorithm is iterative.

3.3 Direct ellipse-specific fitting

In order to fit ellipses specifically while retaining the efficiency of solution of the linear least-squares problem (2.2), we would like to constrain the parameter vector $\mathbf{a}$ so that the conic that it represents is forced to be an ellipse. The appropriate constraint is well known, namely that the *discriminant* $b^2 - 4ac$ be negative. However, this constrained problem is difficult to solve in general as the Kuhn-Tucker conditions [99] do not guarantee a solution. In fact, I have not been able to locate any reference regarding the minimization of a quadratic form subject to such a nonconvex inequality.

Although imposition of this inequality constraint is difficult in general, in this case we have the freedom to arbitrarily scale the parameters so we may simply incorporate the scaling into the constraint and impose the *equality* constraint $4ac - b^2 = 1$. This is a quadratic constraint which may be expressed in the matrix form $\mathbf{a}^T\mathbf{C}\mathbf{a} = 1$ as

$$\mathbf{a}^T \begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{a} = 1 \tag{3.1}$$

3.3.1 Solution of the quadratically constrained minimization

Before we approach the minimization problem, we note that the standard form of the conic

$$F(\mathbf{a}; \mathbf{x}) = ax^2 + bxy + cy^2 + dx + ey + f = 0$$

may be written in a way that separates the parameters $\mathbf{a}$ from the terms in $\mathbf{x}$ using the dot product

$$\begin{aligned} F(\mathbf{a}; \mathbf{x}_i) &= [x_i^2 \; x_i y_i \; y_i^2 \; x_i \; y_i \; 1] \cdot [a \; b \; c \; d \; e \; f] \\ &= \chi_i \cdot \mathbf{a} \end{aligned}$$

Where χ_i is the *design vector* of the point (x_i, y_i). We then assemble these design vectors into the $n \times 6$ *design matrix* $\mathbf{D}$:

$$\mathbf{D} = \begin{pmatrix} x_1^2 & x_1 y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2 y_2 & y_2^2 & x_2 & y_2 & 1 \\ & & \cdots & & & \\ x_n^2 & x_n y_n & y_n^2 & x_n & y_n & 1 \end{pmatrix}$$

Following Bookstein [16], the constrained fitting problem is then:

$$\text{Minimize } E = \|\mathbf{Da}\|^2, \text{ subject to the constraint } \mathbf{a}^T\mathbf{Ca} = 1 \tag{3.2}$$

Introducing the Lagrange multiplier λ and differentiating we arrive at the system of simultaneous equations[1]

$$\boxed{\begin{aligned} 2\mathbf{D}^T\mathbf{Da} - 2\lambda\mathbf{Ca} &= 0 \\ \mathbf{a}^T\mathbf{Ca} &= 1 \end{aligned}} \tag{3.3}$$

This may be rewritten as the system

$$\mathbf{Sa} = \lambda\mathbf{Ca} \tag{3.4}$$

$$\mathbf{a}^T\mathbf{Ca} = 1 \tag{3.5}$$

where $\mathbf{S}$ is the *scatter matrix* $\mathbf{D}^T\mathbf{D}$. To solve this system of equations, we first get all possible solutions to (3.4), and then select those which are consistent with (3.5). To get all solutions of (3.4) we consider its generalized eigenvectors: the set $\{\lambda_i, \mathbf{u}_i\}_{i=1}^6$ such that

$$\mathbf{Su}_i = \lambda_i\mathbf{Cu}_i \tag{3.6}$$

$$\|\mathbf{u}_i\| = 1 \tag{3.7}$$

If $(\lambda_i, \mathbf{u}_i)$ solves (3.4) then so does $(\lambda_i, \mu_i\mathbf{u}_i)$ for any μ_i. Therefore any solution $\mathbf{a}$ must be a scalar multiple $\mathbf{a} = \mu_i\mathbf{u}_i$ of one of the generalized eigenvectors. By applying the constraint (3.5) we get

$$\mu_i^2\mathbf{u}_i^T\mathbf{Cu}_i = 1$$

giving

$$\begin{aligned} \mu_i &= \sqrt{\frac{1}{\mathbf{u}_i^T\mathbf{Cu}_i}} \\ &= \sqrt{\frac{\lambda_i}{\mathbf{u}_i^T\mathbf{Su}_i}} \end{aligned} \tag{3.8}$$

Finally, setting $\hat{\mathbf{a}}_i = \mu_i\mathbf{u}_i$ solves (3.3). As in general there may be up to 6 real solutions, the solution is chosen that yields the lowest residual $\hat{\mathbf{a}}_i^T S\hat{\mathbf{a}}_i = \lambda_i$.

We note that the solution of the eigensystem (3.4) gives 6 eigenvalue-eigenvector pairs $(\lambda_i, \mathbf{u}_i)$. Each of these pairs gives rise to a local minimum if the term under the square root in (3.8) is positive. In general, $\mathbf{S}$ is positive definite, so the denominator $\mathbf{u}_i^T\mathbf{Su}_i$ is positive for all $\mathbf{u}_i$. Therefore the square root exists if $\lambda_i > 0$, so any solutions to (3.3) must have positive generalized eigenvalues.

3.3.2 Analysis of the constraint $4ac - b^2 = 1$

Now we show that the minimization of $\|\mathbf{Da}\|^2$ subject to $4ac - b^2 = 1$ yields exactly one solution (which corresponds, by virtue of the constraint, to an ellipse). For the demonstration, we will require the following lemma (proved in Appendix A):

Lemma 1 *The* signs *of the generalized eigenvalues of* $\mathbf{Su} = \lambda\mathbf{Cu}$ *are the same as those of the constraint matrix* $\mathbf{C}$*, up to permutation of the indices.*

Theorem 1 *The solution of the constrained optimization problem (3.2) admits exactly one elliptical solution corresponding to the single positive generalized eigenvalue of (3.4).*

[1] Note that the method of Lagrange multipliers is not valid when the gradient of the constraint function becomes zero. In (3.2) this means $\mathbf{Ca} = 0$, but then $\mathbf{a}^T\mathbf{Ca} = 0$ so the constraint is violated and there is no solution.

Proof:

Since the eigenvalues of $\mathbf{C}$ are $\{-2, -1, 2, 0, 0, 0\}$, from Lemma 1 we have that (3.4) has exactly one positive eigenvalue $\lambda_i > 0$, giving the unique solution $\hat{\mathbf{a}} = \mu_i \mathbf{u}_i$ to (3.3). When $\mathbf{D}^T\mathbf{D}$ is positive definite, all local minima of the constrained problem must satisfy (3.3), and we conclude that $\hat{\mathbf{a}}$ solves the constrained problem. In the case where $\mathbf{D}^T\mathbf{D}$ is positive semidefinite (which occurs with degenerate data), there are infinitely many equivalent minima including the given solution. □

3.4 Observations on the Ellipse-specific constraint

In this section I make a number of observations on the properties of the ellipse-specific constraint.

3.4.1 Constraint as normalization

An eigenvector of the eigensystem (3.4) is a local minimizer of the *Rayleigh quotient* $\frac{\mathbf{a}^T\mathbf{S}\mathbf{a}}{\mathbf{a}^T\mathbf{C}\mathbf{a}}$. In this case we may think of the constraint $4ac - b^2 = 1$ as a normalization factor on the error-of-fit function, thus considering the error norm

$$EOF = \sum_{i=1}^{n} \frac{ax^2 + bxy + cy^2 + dx + ey + f}{4ac - b^2}$$

Rosin [104] writes that, not surprisingly, the minimization tends to "pull" the solution away from singularities; in the ellipse-specific case the singularity is a parabola and so the unique elliptical solution tends to be biased towards low eccentricity.

3.4.2 Geometric interpretation

An interesting geometric interpretation of the effect of the ellipse normalization comes from noting that the discriminant is proportional to the squared inverse of the ellipse's area. That is

$$4ac - b^2 \propto \frac{1}{(R_x R_y)^2}$$

Hence we note that the division by $4ac - b^2$ is equivalent to multiplication by the ellipse area. Interestingly, this is the same weighting used by Solina and Bajcsy when fitting superquadrics!

3.5 Experimental Results

In this section we present experimental results that compare the ellipse-specific solution to previous methods in terms of quality and robustness. Both quantitative and qualitative results are included in order to allow other researchers to evaluate the utility of the ellipse-specific algorithm with respect to the others cited.

3.5.1 Ellipse-specificity

Despite the theoretical proof of the algorithm's ellipse-specificity, it is instructive to observe its performance on some example data, of which Figure 3.1 provides an example. There, the output of Taubin's method and the proposed method are shown for the same set of data. Taubin's algorithm gives a hyperbola as the best solution which, while an accurate representation of the data, is of little use if ellipses are sought. In contrast, the ellipse-specific algorithm returns an ellipse as expected.

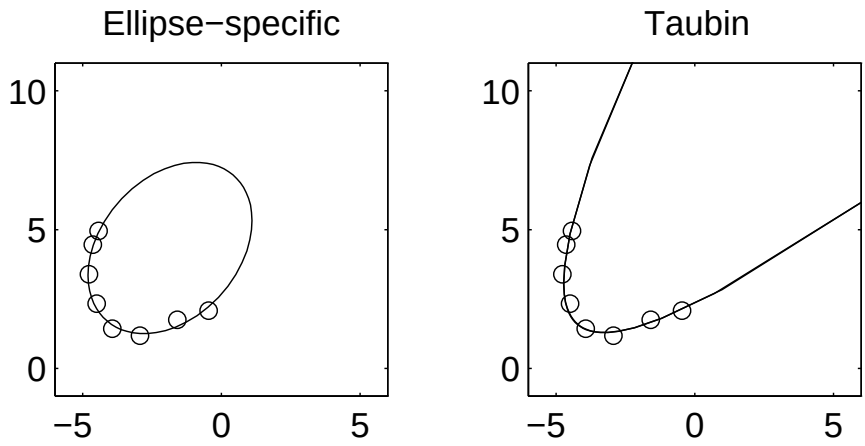


Figure 3.1: A simple demonstration of the algorithm's ellipse specificity. For the hand-drawn example data, the Taubin algorithm returns a hyperbola, while the new algorithm returns an ellipse.

3.5.2 Experimental procedure

The experimental procedure for generating an ellipse with known ground truth is given in this section. Although the experiment uses only one ellipse, the more detailed experiments in the next chapter prove that the qualitative results from this case extend to the general case.

1. Generate 100 points from the ellipse $0.2678x^2 + 0.9822y^2 - 0.2286xy = 1$, which has $R_x = 2, R_y = 1$ and is rotated 30° anticlockwise from the positive X axis. The points are generated from

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} R_x * \cos(\theta) \\ R_y * \sin(\theta) \end{pmatrix}$$

   for θ varying linearly between -70° and $+70^\circ$. Although this will not produce unit arc-length samples, in this case the eccentricity is small enough that the bias this introduces does not significantly affect the results.

2. Corrupt each point by adding zero-mean Gaussian noise of standard deviation $\sigma = N/40$, where N are the noise levels cited under the figures. The reason for division by 40 is to make the 8×6 axes in which the figures are presented correspond more intuitively to a real image size of 320×240. The noise levels may then be read in pixels rather than in the normalized units used here.

3.5.3 Qualitative noise sensitivity

Now we qualitatively assess the robustness of the method to noise and compare it to the Linear, Taubin, Bookstein and $a + c = 1$ algorithms. The experimental approach explores the repeatability of the algorithms when presented with many data sets, each draw from the same noise population, but with difference instances of the noise. The algorithms may then be rated on *accuracy*: how closely the average returned ellipse matches the true ellipse; and *stability*: how tightly the returned ellipses cluster around the average.

Figure 3.2, and the following figures, superimpose the conics returned by each algorithm on one display, allowing a visual evaluation of the central tendency and spread in

each case. The primary results are that Taubin's algorithm is more accurate at low noise levels, but that the ellipse-specific algorithm is more stable at all levels, and is most accurate at all levels other than the low noise case, where it is second most accurate.

3.5.4 Geometric Distance Error

The second noise experiment measures the average geometric distance error for each of the algorithms over 100 runs. In order to verify that the ellipses returned by the new algorithm are reasonable approximations to the minimum geometric distance ellipse, non-elliptical fits returned by the Bookstein and Taubin algorithms were ignored. In Figure 3.8, it can be seen that the new algorithm produces a closer ellipse on average than Bookstein's for medium noise, but that Taubin's—when it returns an ellipse—produces the smallest geometric distance error. We note however that all results are within each other's 1σ error bars over the 100 runs, meaning that the variations within runs are greater than the difference between the algorithms across runs.

3.5.5 Affine transformation invariance

The quadratic constraint proposed not only constrains the fitted conics to be ellipses but is also rotation and translation invariant, as shown in Appendix C. As an experimental verification, in two sets of experiments a randomly rotated and translated data set was created, and for each fit the recovered parameters were compared to the expected ones. In both experiments the difference between expected semi-axes, centre position and rotation was zero up to machine precision.

3.6 Conclusions

This chapter has presented a new method for direct least square fitting of ellipses. This is believed to be the first noniterative ellipse-specific algorithm. Previous conic fitting methods rely (when applied to ellipse fitting) either on the presence of good data or on computationally expensive iterative updates of the parameters.

A theoretical demonstration shows that the new method uniquely yields elliptical solutions which, under the normalization $4ac - b^2 = 1$, minimize the sum of squared algebraic distances from the points to the ellipse. In addition, a theoretical demonstration of affine invariance is supported by numerical tests.

The experimental results may be summarized as follows: under high noise or occlusion, the new algorithm is the most robust of all presented. Under low noise, the best tradeoff between speed and accuracy is probably given by the Taubin algorithm, with a fallback to the new algorithm in the cases where the Taubin method fails to return an ellipse. For initialization of nonlinear algorithms, the new algorithm is the method of choice, being quick and robust. The stability properties widen the scope of application of the algorithm from ellipse fitting to cases where the data are not strictly elliptical, but need to be minimally represented by an elliptical "blob".

The simplicity of the algorithm is demonstrated by the inclusion in Fig.3.9 of a complete 6-line implementation in MATLAB. In the further interests of dissemination, an implementation of the algorithm is included in releases of the Image Understanding Environment (IUE) [79], versions 1.3 and above.

We note also that the algorithm can be trivially converted to a hyperbola-specific fitter (simply change the sign of the constraint).

The algorithm is however biased towards ellipses of low eccentricity, and future work includes the incorporation of the algorithm into a bias-correction algorithm based on that of Kanatani [55]. In a similar vein, a theoretical analysis of the noise performance of the methods using eigensystem perturbation theory is under investigation.

Figure 3.2:

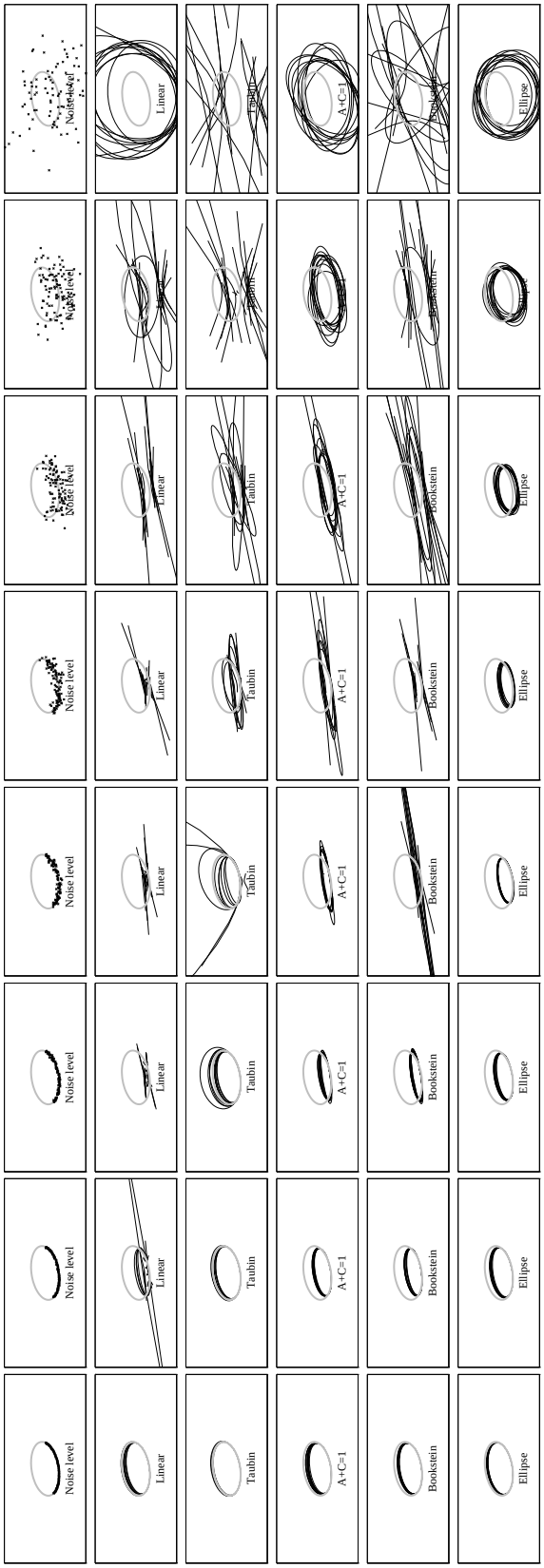


Figure 3.2. Overview of qualitative results. Noise is increasing from left to right and the algorithms are (from top to bottom): Linear, Taubin, A+C=1, Bookstein, Ellipse-specific. Expanded figures appear in the following pages.

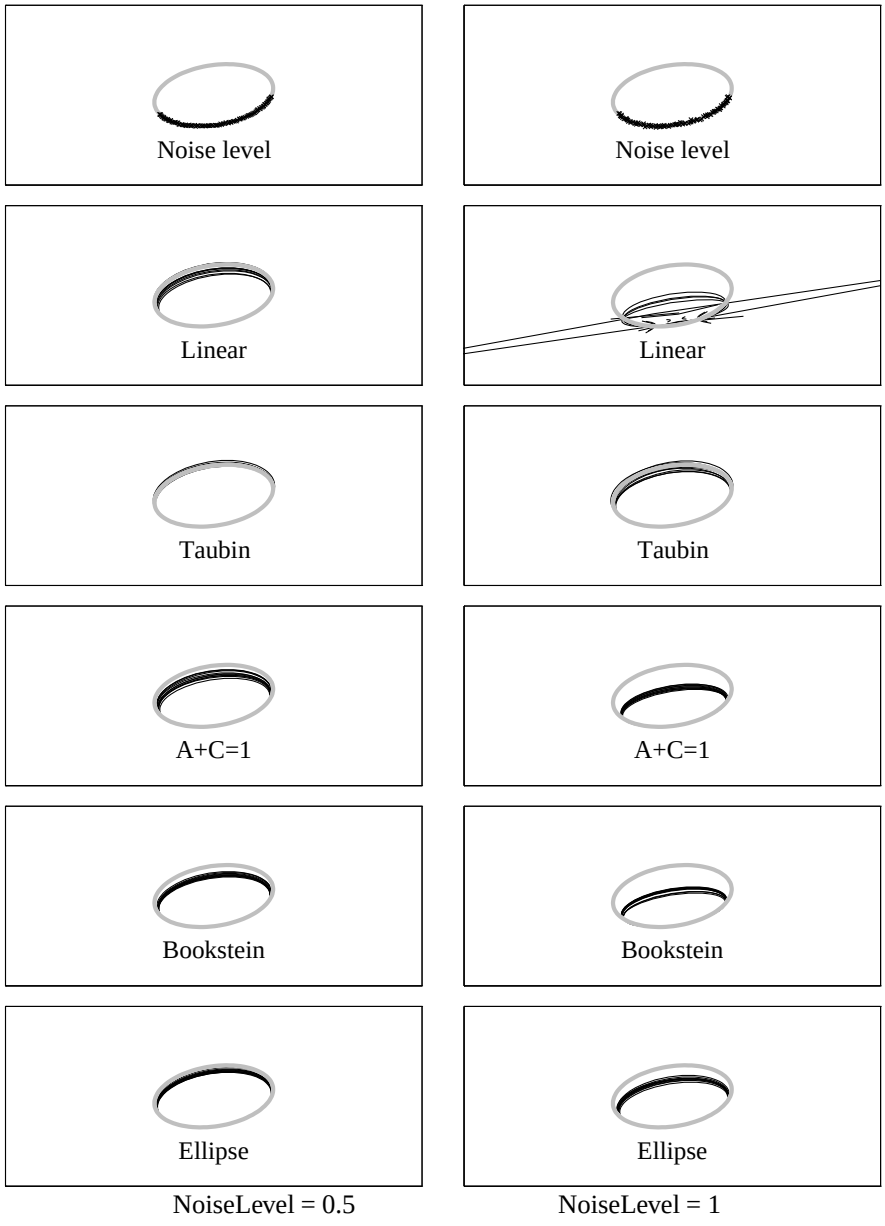


Figure 3.3: Qualitative performance for low noise. In the lowest noise case, Taubin's constraint gives the best results: the returned conics are all ellipses, and the ellipses are clustered about the ground truth (shown in grey). The next most accurate algorithm is the proposed ellipse-specific one.

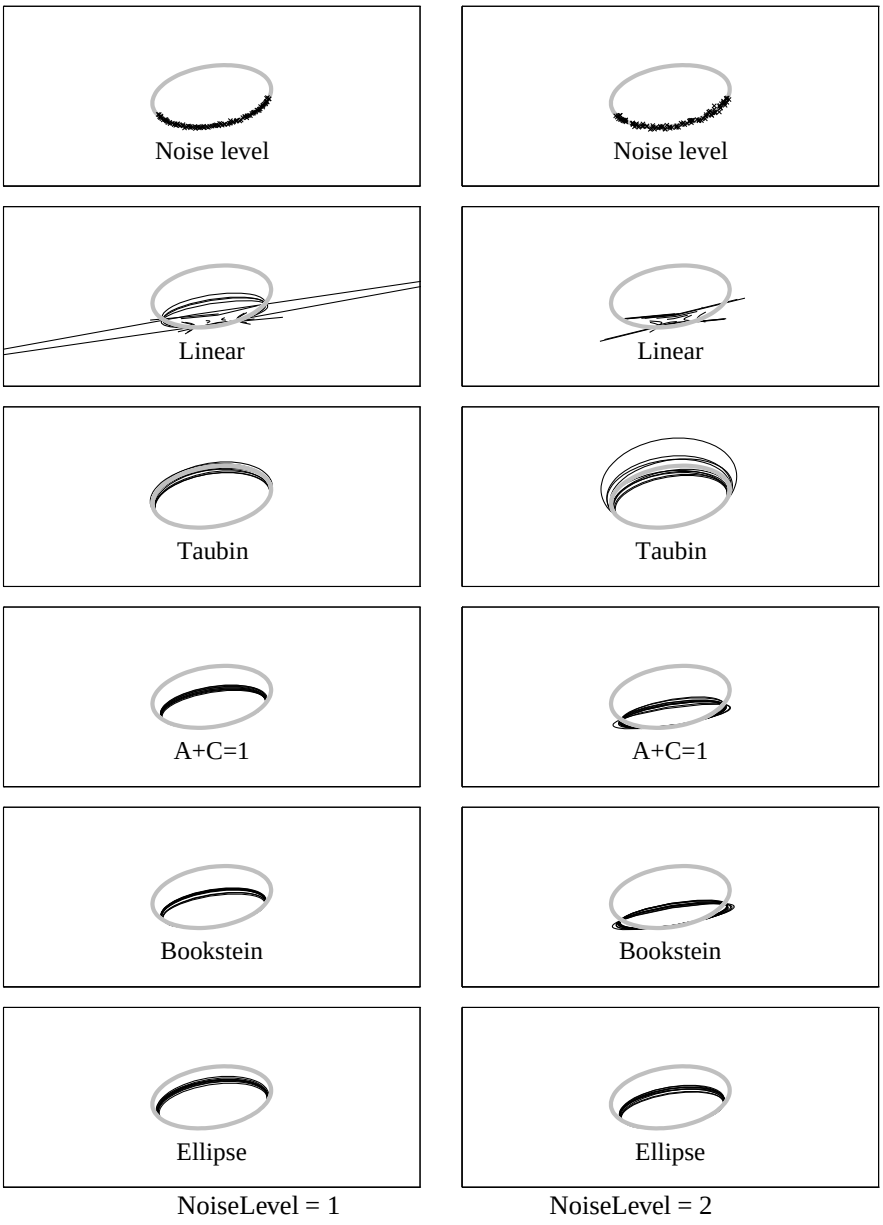


Figure 3.4: Qualitative performance for medium noise (note that column 1 is the same as column 2 of the previous page to allow comparison). Taubin remains the most accurate on average, but its variance is higher than the ellipse-specific. Linear begins to break down.

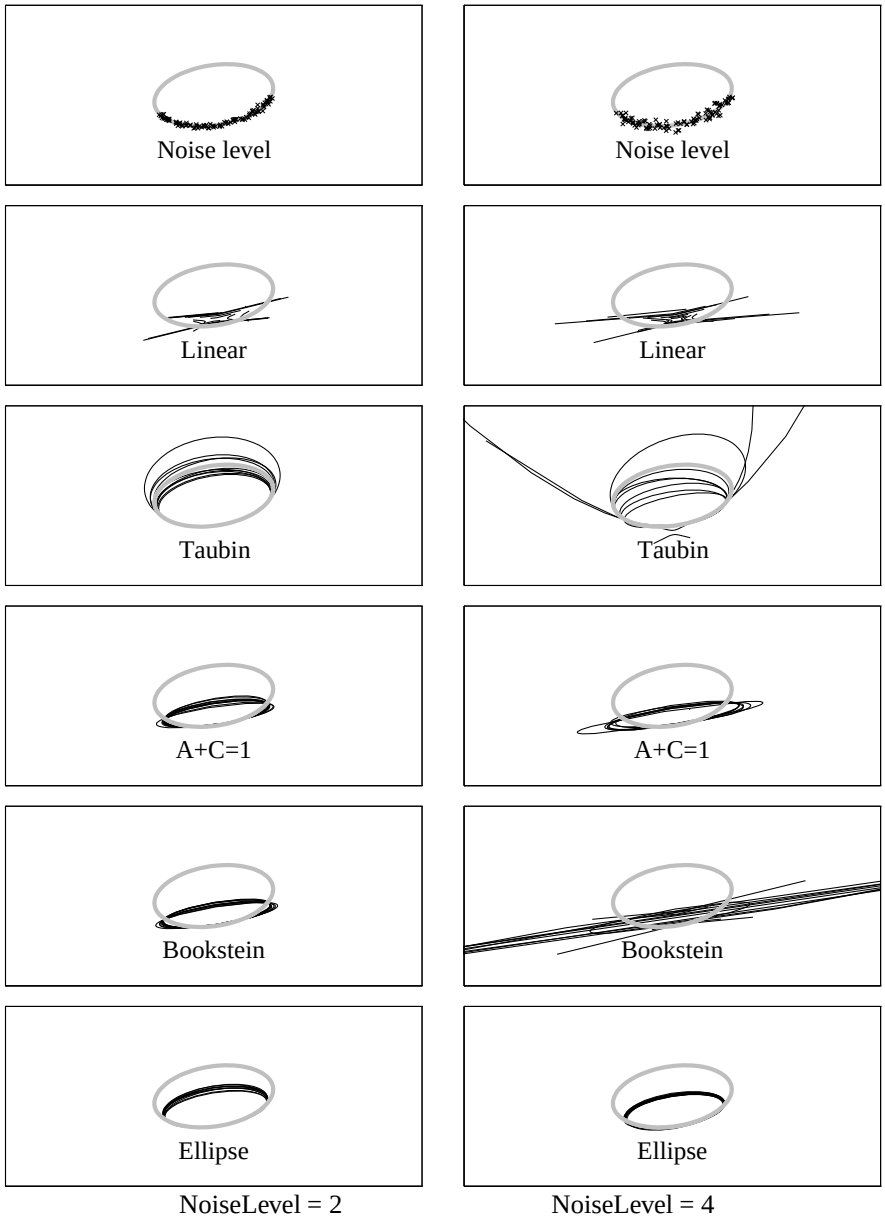


Figure 3.5: Qualitative performance for medium-to-high noise. Bookstein breaks down, $a + c = 1$ begins to do so.

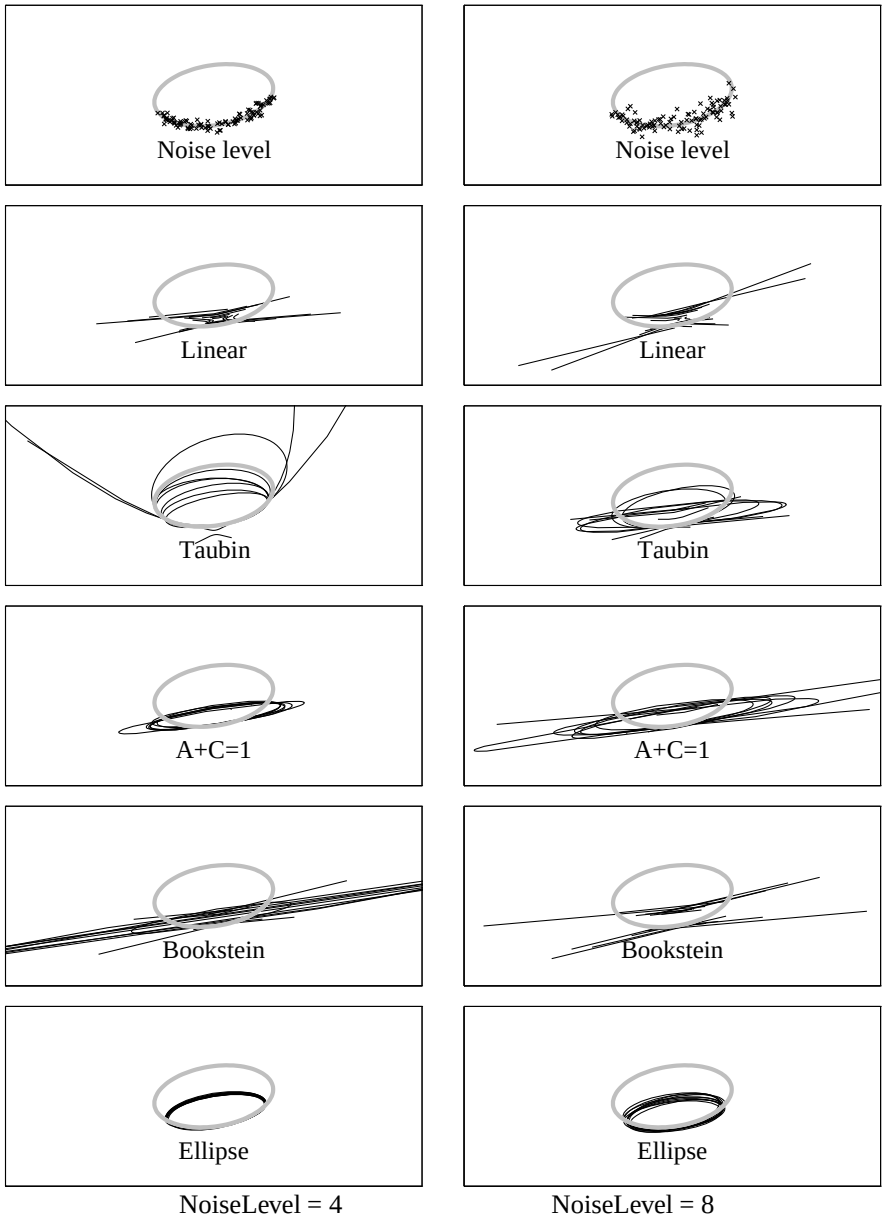


Figure 3.6: Qualitative performance for high noise. All other algorithms break down. The ellipse-specific algorithm loses accuracy and some stability but remains stable.

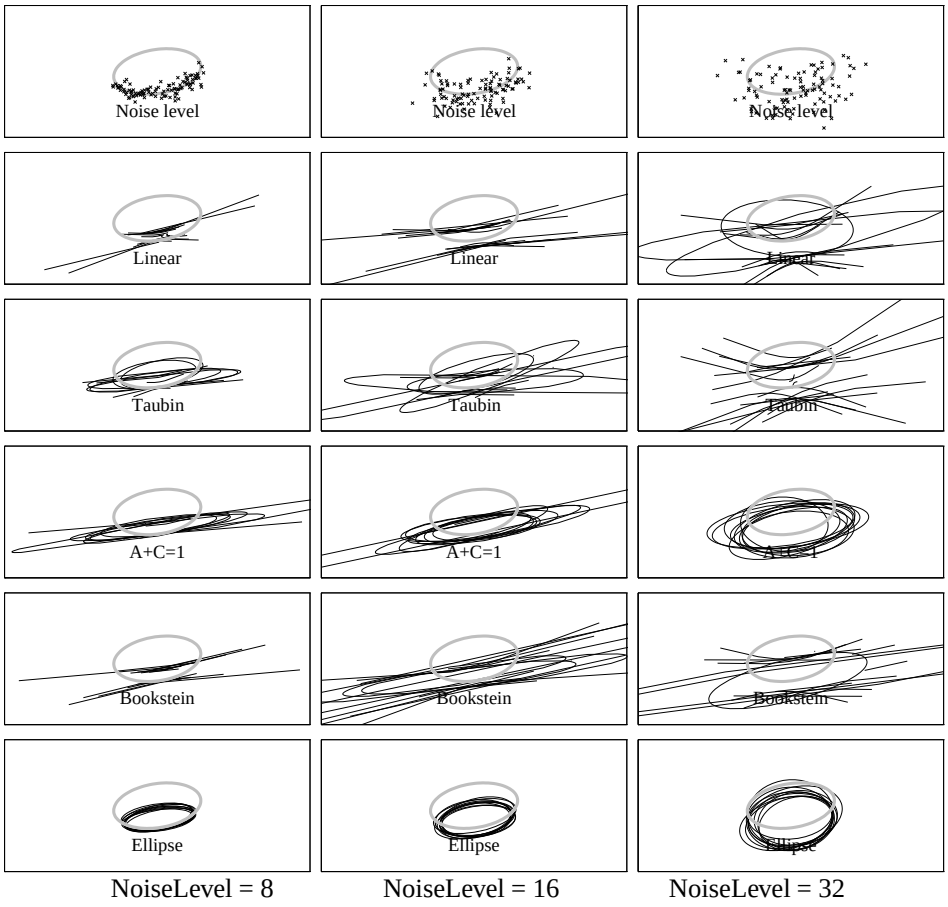


Figure 3.7: Qualitative performance for increasingly unlikely amounts of noise. The ellipse-specific algorithm demonstrates its instability by degrading to become an estimator of central tendency.

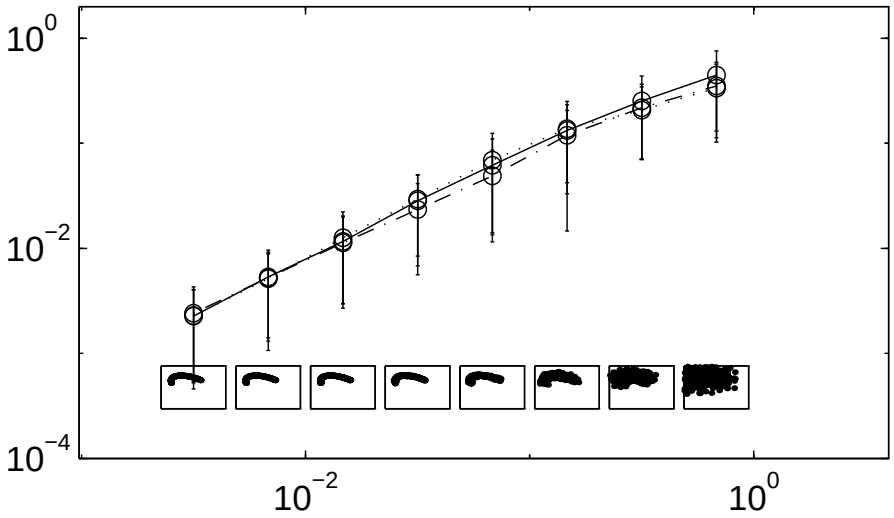


Figure 3.8: Average geometric distance error as a function of increasing noise level. The errorbars are at $\pm 1\sigma$. The pictures along the noise axis indicate visually the corresponding noise level of the points fitted. Encoding is Bookstein: dotted; Taubin: dash-dot; New: solid.

```
% x,y are lists of coordinates
function a = fit_ellipse(x,y)
% 1. Build design matrix
D = [ x.*x x.*y y.*y x y ones(size(x)) ];
% 2a. Build scatter matrix
S = D'*D;
% 2b. Build 6x6 constraint matrix
C(6,6) = 0; C(1,3) = 2; C(2,2) = -1; C(3,1) = 2;
% 3a. Solve eigensystem
[gevec, geval] = eig(inv(S)*C);
% 3b. Find the positive eigenvalue
[PosR, PosC] = find(geval > 0 & ~isinf(geval));
% 3c. Extract eigenvector corresponding to positive eigenvalue
a = gevec(:,PosC);
```

Figure 3.9: Complete 6-line Matlab implementation of the proposed algorithm.

Chapter 4

Experiments in Conic Fitting

The previous chapter has sketched out the importance of ellipse fitting in computer vision — this chapter approaches the more common topic of conic fitting in order to experimentally verify the noise and occlusion stability of the ellipse specific algorithm. Vision applications often require the extraction of conic sections from image data. Common examples are the calculation of geometric invariants [39] and estimation of the centers and radii of circles for industrial inspection. Camera calibration algorithms often require the extraction of circles or hyperbolae [26].

4.1 Introduction

Many textbooks [49, 93] provide discussions and algorithms for least-squares approximation of conics, but these often include only the simple and fast algebraic distance algorithm (algorithm "Linear" below). This algorithm fares poorly on many real data sets due to its inherent statistical bias, particularly when the image curves are partially occluded. A number of authors [16, 55, 80, 91, 115, 124] have proposed alternative algorithms and while these are usually compared by the authors with the linear algorithm, there have been, to my knowledge, no comparative empirical studies of the relative accuracy and efficiency of these alternatives, although a theoretical comparison of distance metrics is provided by [104, 105].

This chapter makes two important contributions to this area of computer vision research:

- Identification of the main conditions under which the algorithms fail. It is common for comparative evaluations to concentrate on noise sensitivity, but in the case of conic fitting the important parameter is the amount of occlusion.
- Presentation of the algorithm computational costs in terms of *flop* counts (see §4.3.1) allows evaluation of the tradeoff between accuracy and speed of execution without reference to the specifics of an implementation and environment.

The methods compared differ primarily in terms of the error measure that they minimize and then in terms of the techniques that are used to minimize this measure. In particular, the error measure determines the eventual accuracy of the methods and generally dictates the choice of optimization algorithm and hence their execution speed. Also, as explained in the conclusion, I concentrate largely on "one-shot" algorithms which are known to return the global minimum of the distance metric, rather than iterative minimization techniques.

I begin with a brief summary of the implementation and costs of each of the algorithms. The broader issue of experimental methodology is then addressed, considering the issues of generation of synthetic data sets, choice of metric for comparison, and choice of parameters to characterize performance. Finally, the experiments themselves are presented, and

conclusions drawn as to the conditions under which each algorithm should be used. Emphasis is laid on the design of experiments to test the algorithms fully, rather than reliance on intuition.

4.2 Problem Statement

The problem that the algorithms presented in this chapter solve may be stated as follows. In the notation of Chapter 1, the curve families that we use are the conics, for which

$$F(\mathbf{a};\mathbf{x}) = ax^2 + bxy + cy^2 + dx + ey + f$$

and circles, for which

$$F_C(\mathbf{a};\mathbf{x}) = (x^2 + y^2) + ax + by + c$$

and the algorithms find the $\mathbf{a}$ that minimizes an error function $\epsilon^2(\mathbf{a})$, possibly subject to some constraint on the domain of $\mathbf{a}$.

4.3 The Algorithms

Previous sections have reviewed the conic fitters at a computational and conceptual level. In this section, I more precisely summarize the algorithmic level, in order to document the precise implementations that have been used to generate the results in this chapter.

4.3.1 Flop counts

Computational costs of the algorithms below are presented in terms of *flop* counts [43]. This is a method of providing a complexity measure that is independent of the hardware and operating environment under which experiments are performed. A *flop* comprises a single "**fl**oating point **op**eration", such as a multiplication or addition, and includes the concomitant overheads of subscripting and memory access. However, the precise definition of what constitutes a single flop can vary among authors. Moler's MATLAB system [75], for example, defines addition and multiplication to each contribute one flop, while Golub and van Loan consider one flop to consist of a multiply-accumulate. The MATLAB definition corresponds more closely to the computer used to perform the experiments, on which both multiplication and addition require one clock cycle, and to most modern computers. For computers which exhibit a large variation between the costs of multiplication and addition, the values provided here will overestimate the complexity of the algorithms. However, the relative costs will almost certainly remain similar. Table 4.1 summarizes the costs of the algorithms

4.3.2 Algorithm "Linear": Algebraic Distance

The algebraic distance algorithm minimizes the objective function

$$\epsilon^2(\mathbf{a}) = \sum_{i=1}^{n} F(\mathbf{a},\mathbf{x}_i)^2 = \|\mathbf{D}\mathbf{a}\|^2$$

subject to the constraint that $\|\mathbf{a}\|^2 = 1$. The *design matrix* $\mathbf{D}$ is the $n \times 6$ matrix with rows χ_i introduced in the previous chapter. The constrained objective function $E = \|\mathbf{D}\mathbf{a}\|^2 - \lambda(\|\mathbf{a}\| - 1) = \mathbf{a}^\top\mathbf{D}^\top\mathbf{D}\mathbf{a} - \lambda(\mathbf{a}^\top\mathbf{a} - 1)$ is minimized analytically to form an eigenvector problem [43]:

$$\nabla_{\mathbf{a}} E = 0 \quad \Longleftrightarrow \quad 2\mathbf{D}^\top\mathbf{D}\mathbf{a} - 2\lambda\mathbf{a} = 0$$

Algorithm	Cost in flops	Direct/Iterative
A+C=1	26 n + 155	D
Linear	26 n + 1700	D
Bookstein	26 n + 1165	D
Ellipse	26 n + 5182	D
Taubin	26 n + 9700	D
Kanatani	$\approx$ 500 n	I
Sampson	$\approx$ 5000 n	I
Geometric	$\approx$ 10000 n	I

Table 4.1: Summary of computational costs of the algorithms

where λ is a Lagrange multiplier. The minimizer $\mathbf{a}_{\min}$ is then the eigenvector of $\mathbf{D}^\top\mathbf{D}$ corresponding to the smallest eigenvalue. (As $\mathbf{D}^\top\mathbf{D}$ is positive definite, all the eigenvalues are positive. If $\mathbf{D}^\top\mathbf{D}$ is not positive definite, this signifies that the problem is poorly conditioned—see §4.5).

The algorithm requires $12n$ multiplications and $14n$ adds (or approximately $26n$ *flops*) for the construction of the 15 unique elements of $\mathbf{D}^\top\mathbf{D}$. Evaluation of the eigensystem by Hessenberg reduction and QR generally took about 20 iterations (1700 flops) in these experiments, giving a total cost of about $26n + 1700$ flops.

4.3.3 Algorithm "Bookstein": Algebraic distance with quadratic constraint

Bookstein's algorithm [16] attempts to reduce the bias of the linear algorithm by imposing a different constraint on the parameter vector $\mathbf{a}$. He derives the constraint $a^2 + \frac{b^2}{2} + c^2 = 1$ by consideration of the invariance requirements of fitting, and shows how this leads to the system

$$\mathbf{D}^\top\mathbf{D}\mathbf{a} = \lambda\,\Delta\mathbf{a}$$

where $\Delta = \mathrm{diag}(2, 1, 2, 0, 0, 0)$. This is a rank-deficient generalized eigensystem, which Bookstein solves by block-decomposition.

Bookstein requires the $26n$ flops of "Linear" to form the $\mathbf{D}^\top\mathbf{D}$ matrix. The matrix inversion and eigensystem solution's mean flop count was 1165, yielding a total cost of $26n + 1165$ flops.

4.3.4 Algorithm "A+C=1": Algebraic distance with linear constraint

The $a + c = 1$ algorithm is popular in many applications [91, 108, 104, 40], possibly because of the slight bias towards ellipses that it shows in the experiments of this chapter. The constraint is implemented by solving the system

$$\begin{array}{ccc} \mathbf{M} & & \mathbf{a}' \\ \begin{pmatrix} x_1y_1 & y_1^2 - x_1^2 & x_1 & y_1 & 1 \\ x_2y_2 & y_2^2 - x_2^2 & x_2 & y_2 & 1 \\ & & \vdots & & \\ x_ny_n & y_n^2 - x_n^2 & x_n & y_n & 1 \end{pmatrix} & \begin{pmatrix} b \\ c \\ d \\ e \\ f \end{pmatrix} & = \begin{pmatrix} -x_1^2 \\ -x_2^2 \\ \vdots \\ -x_n^2 \end{pmatrix} \end{array} = \mathbf{b}$$

Computing $a = 1 - c$ then gives the 6-parameter conic vector. The algorithm as stated requires $4n$ flops to form the design matrix, and about $100n$ flops to solve the linear system by QR decomposition, but it can be implemented as a 5×5 linear system

$$\mathbf{M}^\top\mathbf{M}\mathbf{a}' = \mathbf{M}^\top\mathbf{b}$$

which is solved by LU decomposition. In this case the total cost is just $26n + 155$ flops, making $a + c = 1$ the cheapest of the algorithms.

4.3.5 Algorithm "Taubin": Approximate mean square distance

The "approximate mean square distance" (AMS) metric, introduced by Taubin [124], minimizes the unusual objective function

$$\epsilon^2(\mathbf{a}) = \frac{\sum_{i=1}^{n} F(\mathbf{a}, \mathbf{x}_i)^2}{\sum_{i=1}^{n} \|\nabla_{\mathbf{x}} F(\mathbf{a}, \mathbf{x}_i)\|^2} = \frac{\|\mathbf{D}\mathbf{a}\|^2}{\|\mathbf{D}_x\mathbf{a}\|^2 + \|\mathbf{D}_y\mathbf{a}\|^2}$$

where the matrices $\mathbf{D}_x$ and $\mathbf{D}_y$ are the partial derivatives of $\mathbf{D}$ with respect to x and y. Restating the problem as the minimization of $\|\mathbf{D}\mathbf{a}\|^2$ subject to $\|\mathbf{D}_x\mathbf{a}\|^2 + \|\mathbf{D}_y\mathbf{a}\|^2 = 1$, the minimizer $\mathbf{a}_{\min}$ is then the eigenvector of the generalized eigensystem [43]

$$\mathbf{D}^\top\mathbf{D}\mathbf{a} = \lambda(\mathbf{D}_x^\top\mathbf{D}_x + \mathbf{D}_y^\top\mathbf{D}_y)\mathbf{a}$$

corresponding to the smallest eigenvalue.

Taubin requires the $26n$ flops of the linear algorithm to form the $\mathbf{D}^\top\mathbf{D}$ matrix, but negligible additional time to form $\mathbf{D}_x^\top\mathbf{D}_x + \mathbf{D}_y^\top\mathbf{D}_y$ from the elements of $\mathbf{D}$. The generalized eigensystem routine's mean flop count was 9700, yielding a total cost of $26n + 9700$ flops.

4.3.6 Algorithm "Kanatani": Statistical distance

Another approach, proposed by Kanatani [55] and Porrill [91], improves the linear algorithm by explicitly calculating its inherent statistical bias and subtracting the bias from the result of minimization. Because the calculation of the bias depends on knowing both the true minimum and the noise level, the process is iterated until the predicted bias results in a noise-level correction of zero. Kanatani calls this metric the "statistical distance", and argues that its bias sensitivity is in fact superior to the geometric distance in the case where the errors on the data points are spherically distributed. This chapter discusses only Kanatani's bias correction algorithm. A description of the algorithm itself would be too involved to include here, due to its dependence on tensor arithmetic. Note, however, that on pursuing the derivation, it is easily discovered that the published noise-level update formula [55, eq 21] should be replaced by

$$c \;\leftarrow\; c + \frac{\lambda_m}{\mathrm{tr}(Q)\mathrm{tr}(MQ) + 2\mathrm{tr}(MQ^2)}$$

Complexity of the algorithm is of the order of $50n + 1000$ flops per iteration, with test runs taking an average of 10 iterations.

4.3.7 Algorithm "Ellipse": Ellipse-specific algebraic distance

The Ellipse algorithm is the ellipse-specific algorithm of the previous chapter, which imposes the constraint that $4ac - b^2 = 1$. This converts the inequality $b^2 - 4ac < 0$ into an equality by incorporating the normalization factor. Ellipse requires the $26n$ flops of the linear algorithm to form the $\mathbf{D}^\top\mathbf{D}$ matrix. The matrix inversion and eigensystem solution's mean flop count was 5182.

4.3.8 Algorithm "Geometric": Geometric Distance

The geometric distance metric measures the orthogonal distance from the point $\mathbf{x}$ to the conic section. This metric, proposed by Nakagawa and Rosenfeld [80], is approximately unbiased if the errors in the data points are distributed normally to the curve, and if each

data point is the image of only one curve point. The distance is evaluated at a point $\mathbf{x}$ by solving the simultaneous equations

$$\begin{aligned} \mathbf{p} + \lambda \nabla F(\mathbf{a}; \mathbf{p}) &= \mathbf{x} \\ F(\mathbf{a}; \mathbf{p}) &= 0 \end{aligned}$$

for $\mathbf{p}$ and defining $\delta(C(\mathbf{a}), \mathbf{x}) = \|\mathbf{x} - \mathbf{p}\|^2$. These equations involve the solution of a quartic equation, and while closed-form solutions exist, numerical instability can result from the application of the analytic formula [93]. In the implementation used here, roots are extracted from the eigensystem of the companion matrix [43]. This, in turn, means that analytic derivatives of δ, and consequently $\nabla_{\mathbf{a}} \epsilon^2(\mathbf{a})$, are difficult to calculate. The Taubin algorithm was used to provide an initial estimate as it minimizes the geometric distance well (see Figure 4.1).

Each evaluation of $\epsilon^2(\mathbf{a})$ involves the solution of n such quartics, averaging $1300n$ flops per iteration for the eigenvalue calculation. The number of iterations depends on the minimization algorithm chosen, but it is clear that even with only 10 iterations, this algorithm is 3 to 4 orders of magnitude slower than the other algorithms considered here, and therefore was not extensively tested in these experiments; being compared only in Figure 4.14.

4.3.9 Algorithm "Sampson": Gradient-reweighted Algebraic Distance

Sampson's refinement of the Bookstein algorithm [115] notes that a first-order approximation to the geometric distance is

$$\epsilon^2(\mathbf{a}) = \sum_{i=1}^{n} \frac{F(\mathbf{a}, \mathbf{x}_i)^2}{\|\nabla_{\mathbf{x}} F(\mathbf{a}, \mathbf{x}_i)\|^2}$$

As direct minimization of this nonquadratic function would require an optimization approach, Sampson's solution was to use a reweighting strategy. At each iteration, he minimizes the modified *weighted least squares* error

$$\epsilon^2(\mathbf{a}) = \sum_{i=1}^{n} w_i F(\mathbf{a}, \mathbf{x}_i)^2$$

where the weights w_i are given by the inverse gradient magnitudes as calculated on the previous iteration. The initial estimate is taken from the Bookstein algorithm, and the Bookstein constraint is applied at each subsequent iteration. The flop count per iteration is therefore that of the Bookstein algorithm added to the $7n$ required to compute the reweight, giving a cost per iteration of $34n$ flops. However this cost is dominated by the need to compute the geometric distance at each iteration—as Sampson notes, his algorithm will not necessarily improve the fit at each iteration, and therefore needs to be monitored using the geometric distance metric. In this sense, Sampson may be considered a faster method of minimizing the geometric distance. In the implementation used here, the algorithm was iterated until the distance converged, averaging about 4 to 10 iterations depending on the occlusion level. The final flop count is then about half of that of the Geometric distance. Again due to its cost, Sampson is considered only in Figure 4.14.

4.4 Experimental Methodology

The primary purpose of the following experiments is to convince the reader that the conclusions drawn as to the relative merits of the methods under examination are reasonable and

accurate. The difficulties inherent in this task are threefold: achieving a thorough exploration of the parameter space; choosing a reasonable method of generating synthetic data; and choosing a meaningful comparison metric.

4.4.1 Exploration of the Parameter Space

The range of parameters on which performance may depend is large [48]. A partial list of those which may affect a conic fitter is as follows:

1. *Noise distribution*: The shape and parameters of the noise distribution may qualitatively affect the algorithm's performances in both relative and absolute terms. For example, we note that in Experiment 1 (see Figure 4.5, page 49), while all the algorithms perform more poorly with increasing noise, the general conic algorithms break down more drastically beyond a certain level.

2. *Occlusion*: As less of the conic is seen, the performance of the algorithms will decrease. The extent of this effect may be different depending both on the amount of occlusion, and also its location with respect to the conic's principal axes. This provides us with two further parameter-space axes along which to attempt to categorize performance.

3. *Position and scale*: A number of the algorithms that are explored in this chapter are not invariant to similarity transformation, and therefore will be expected to behave differently depending on the location, orientation and scale of the data points. Moreover, even algorithms which are theoretically invariant to position may exhibit errors when pixellation effects are included. In this work, I factor out transformation variance by assuming that the data points are transformed to the unit square before fitting, and the pixellation effects may be considered by taking noise level as the scale.

4. *Shape*: The eccentricity of the conic is likely to affect algorithm performance, particularly as it approaches the extremes of lines, circles and degenerate hyperbolae. We note that the affine invariant algorithms will not depend on shape at a coarse scale, but as above, may degrade under extreme imaging conditions.

Parameter Focusing

To provide an intuitive estimate of the size of parameter space that is to be explored in order to cover all possible interactions of these parameters, let us assume that we take 10 noise levels, sample orientation and occlusion in 5° steps, eccentricity at 10 steps and translation at 50. The total number of combinations explored is therefore

$$N_{\text{par}} = 10 \times 72^2 \times 10 \times 72 \times 50^2 \approx 10^{11}$$

It is apparent that to perform even a coarse test of say 100 runs per combination would be prohibitive in terms of compute time. More importantly, presenting the complete results of such a simulation would make it extremely difficult for any general conclusions to be drawn as to the behaviour of the algorithms. However, we want to ensure that any subsampling of this parameter space does not miss any important characteristic of one or more of the algorithms. In an attempt to achieve this sparse—but representative—sampling, I adopt a technique which I have called "parameter focusing".

Parameter focusing attempts to explore the space of possible input data in such a way that parameters which have little qualitative effect on performance are ignored, while a parameter which affects the output is explored in sufficient detail to determine its influence. The level of detail required depends on the effect of the parameter on performance.

- Parameters which affect performance in a cyclic or smoothly varying manner (as does occlusion angle in Experiment 2) need not be sampled densely as their effect will be to smoothly change the performance curves. However, it is useful to set them to their most pessimistic values before exploring other parts of the space, in order to ensure accurate worst-case statistics. Occlusion position is the example in this work, as will be shown throughout the following section.

- Parameters which cause abrupt, or non-smooth, changes in performance should be sampled densely in order to fully illustrate the relative merits of algorithms. The example here is quantity of occlusion.

- Parameters which affect all algorithms equally, simply causing a shift in the performance curves, need not be densely sampled as they will not alter the relative merits of the algorithms.

In order to visualize the space, we are effectively limited to taking one dimensional slices through the space and plotting the value of our chosen error metric. To do this, we must freeze the nonvarying parameters at some representative value. Although the choice of a representative value is intuitively relatively easy in many cases, we would like to be confident that our intuition is supported by experiment. The decision as to what constitutes a representative value depends on the way in which the parameter affects the algorithm, and must be found by taking a dense 1D slice in the direction of that parameter axis, for fixed values of the other parameters. Immediately we see that there is an inherent circularity issue: in order to explore any parameter in order to determine its effect, we need to know how to set the others, but to do so, need to have accurately "sliced" it.

The pragmatic solution to this difficulty is to make parameter focusing an iterative process. An initial feeling for the space can be found using intuition or a sparse sampling of the space (for example random or Latin square sampling [93, p315]). Then the parameters are classified using 1D slicing at sparse intervals, and a more accurate model of the space is built up. The termination criterion for this iteration is that a consistent set of inferences emerges. Then the internal consistency of the final results allows us to infer that if there is a different structure in the parameter space, it is completely orthogonal to the subspaces spanned in the experiments.

As an example, the procedure of this methodology as applied to the experiments in this chapter are as follows:

The initial sparse sampling is simply that gained during testing of the algorithms on many randomly generated problems with all parameters variable. While a more formal approach (such as Latin square sampling) might be preferable, the guided approach appears sufficient for this work.

Initially we explore the effect of noise level on the data. Experiment 1 (see Figure 4.5) illustrates that the results for all algorithms are qualitatively similar with respect to increasing noise. We infer then that the noise axis need not be sampled densely, and experiments are presented for only three levels, qualitatively labelled "low", "medium" and "high".

The effect of occlusion position (the location on the ellipse of the sampled arc center) was initially explored at these three noise levels, revealing that depending on noise level, the worst-case sampling occurred either at curvature maxima or minima.

Examining occlusion reveals quickly that it offers one of the more interesting influences on the algorithms, with Figure 4.10 illustrating the typical behaviour: algorithms perform adequately until a breakdown point is reached, beyond which they fail drastically.

At this point the breakdown behaviour allows us to perform the worst-case search of Experiment 2 in a more informed manner. We conclude that the variation in "worst occlusion position" is in fact dependent on whether algorithms were sampled before or after their breakdown points. The worst case information is of relevance only before breakdown, so we may perform the experiment just once, leading to the more reasonable conclusion that one location is generally worst.

4.4.2 Generation of representative synthetic data

In the current context, we assume that we will be generating synthetic datasets in order to compare the results of fitting with the ground truth, although at the end of the chapter, some examples on real images are provided. In theory this might mean that the entire imaging pipeline ought to be accurately simulated, a relatively difficult task. In this context, however, we may return to the discussion of Kanatani's statistical distance in Section 2.2.4. There it was shown theoretically and empirically that data generated by sampling points from a synthetic curve and adding isotropic Gaussian noise is in fact a reasonable model of the data produced by a real imaging pipeline. This indicates that it is permissible to generate the data for the following experiments by sampling points from a synthetic curve and adding isotropic Gaussian noise.

Other noise models might also be considered, particularly a nonisotropic model where the variance is perpendicular to the curve. However, as the experiments illustrate, degree of occlusion of the curve rather than noise is the parameter to which the algorithms are more sensitive.

Additionally, the noise model does not include any outlier component. This is because none of the described algorithms are statistically robust. The algorithms may be made robust in the usual ways (e.g. RANSAC, or nonlinear minimization using a robust kernel), in which case an outlier component would be added. Qualitatively, the most important change that this might make to the results presented here would be that timing considerations could prove less unfavourable to the iterative algorithms than in the outlier-free case.

Experiments other than the first are performed with three 'typical' noise levels, depicted visually in Figure 4.3. The high, medium and low noise levels correspond roughly to standard deviations of 4, 1 and $\frac{1}{4}$ pixels on an ellipse with a major diameter of 40 pixels. In this and the following, where "pixels" are used as the unit of error measurement, the actual metric used is the level divided by 40, in order to make the results consistent with Figure 4.3.

4.4.3 Choice of comparison metric

A third difficult question lies in the selection of a metric by which to compare the performance of different algorithms. In the ideal case, there is arguably only one truly valid metric. To paraphrase [48]:

> Place each algorithm in the application in which it will be used, run it on typical data (including typical *atypical* data), and compute the running cost (in dollars!) of the overall system.

While this strategy is difficult to refute, it is not useful if one is attempting to provide an indication of the general utility of particular algorithms for typical vision tasks. In order to draw generally applicable conclusions, the choice of error metric is crucial. In addition, Haralick refers primarily to algorithms which produce a discrete output, so that one can count false positives and negatives and give a performance characterization in these terms. In the case of a curve or surface fitting algorithm, the quality of fit is generally a continuous function, so that there is no concept of a "correct" answer – the threshold that will be set is application dependent.

The comparison metric for a single run (i.e. a single instance of corrupted ground truth data) is defined with respect to the ground truth parameters $\mathbf{a}_{\text{true}}$, and the parameters returned by the algorithm for that run $\mathbf{a}_r$ (where r is a subscript which will be used to indicate one of a set of runs used to determine average-case statistics). Then we wish to define the error metric $\mathcal{C}(\mathbf{a}_r, \mathbf{a}_{\text{true}})$ which measures how close the fitted conic is to the ground truth. For the conic fitting case, a number of candidates are immediately apparent:

- Distance between the parameter vectors $\mathcal{C} = \|\mathbf{a}_r - \mathbf{a}_{\text{true}}\|$.

- Angle between the parameter vectors $\mathcal{C} = \mathbf{a}_r \cdot \mathbf{a}_{\text{true}}$. (We assume the parameter vectors have unit norm $\|\mathbf{a}\| = 1$, which is true for the outputs of all the algorithms, even if the fitting constraint used was not $\|\mathbf{a}\| = 1$.)
- Euclidean distance between principal points. Appendix B describes how this measure is computed.
- Euclidean distance between the conic centers (see Appendix B.)
- Difference in estimated radii (see Appendix B.)
- Geometric distance between the fitted conic and the ground truth sample points.

In addition, we can use metrics that are not dependent on knowing the ground truth. For example, if we believe that minimizing the geometric distance gives the MAP estimate, then we may simply measure the geometric distance for each algorithm on the data points and thus compare them. However, as the Kanatani algorithm is based on another definition of MAP estimate, the sum of geometric distances metric will be unfairly biased against the Kanatani algorithm.

On first inspection, the geometrically inspired metrics—principal point distance, difference in radii—are appealing because they correspond to measurable quantities in the image. The metrics based on parameter vectors have no obvious geometric meaning, and the geometric distance favours some algorithms over others. However, in choosing from the geometric measures for general conic fitters, we must note that the center, for example, will often move significantly when the fitters switch from ellipse to hyperbola, while the parameter vectors and geometric distance will remain approximately the same.

In selecting a metric for conic fitters applied to the problem of ellipse fitting, we face a different issue. In this case, we must decide how to deal with algorithms which return non-ellipses without penalizing the ellipse-specific algorithms. In this case, I have presented results for the percentage of non-ellipses returned by each algorithm as a measure of stability. To provide an accuracy measure, I average the value of the comparison metric only for the cases where ellipses are returned.

Figures 4.1 and 4.2 illustrate the performance of these metrics on a low noise dataset. Although I have not yet described the method of producing these results, these figures are introduced now in order to argue that we may choose one as representative for the later experiments. In particular, the units of the abscissa have not been explained, but can be taken to indicate degree of occlusion. The graphs show some variation in their response for different algorithms: for example, the geometric distance error shows a strong bias in favour of the Taubin and Kanatani algorithms, while the "percentage of non-ellipses" metric clearly makes no sense for the ellipse-specific algorithm. However, the overall ordering of the different algorithms remains remarkably constant: The Linear, Bookstein and $a + c = 1$ algorithms form one group which break down earliest; followed by the ellipse-specific algorithm; and the Taubin and Kanatani algorithms tend to break down later. We can therefore use just one of these metrics in the following sections and expect that the conclusions drawn will extrapolate to the others. The particular metric chosen is the principal point distance, because it is the geometric measure with the greatest discriminatory power.

4.5 Experimental Results

All experiments were conducted using the MATLAB system [75]. Eigensystems are solved using the underlying EISPACK routines, while the derivative-free minimization needed for the geometric distance algorithm used the Nelder-Mead simplex algorithm. Also, as the execution-speed characteristics of interpreted MATLAB programs are qualitatively different to those of equivalent programs in a compiled language, we give no timing statistics other than the flop counts in the previous section.

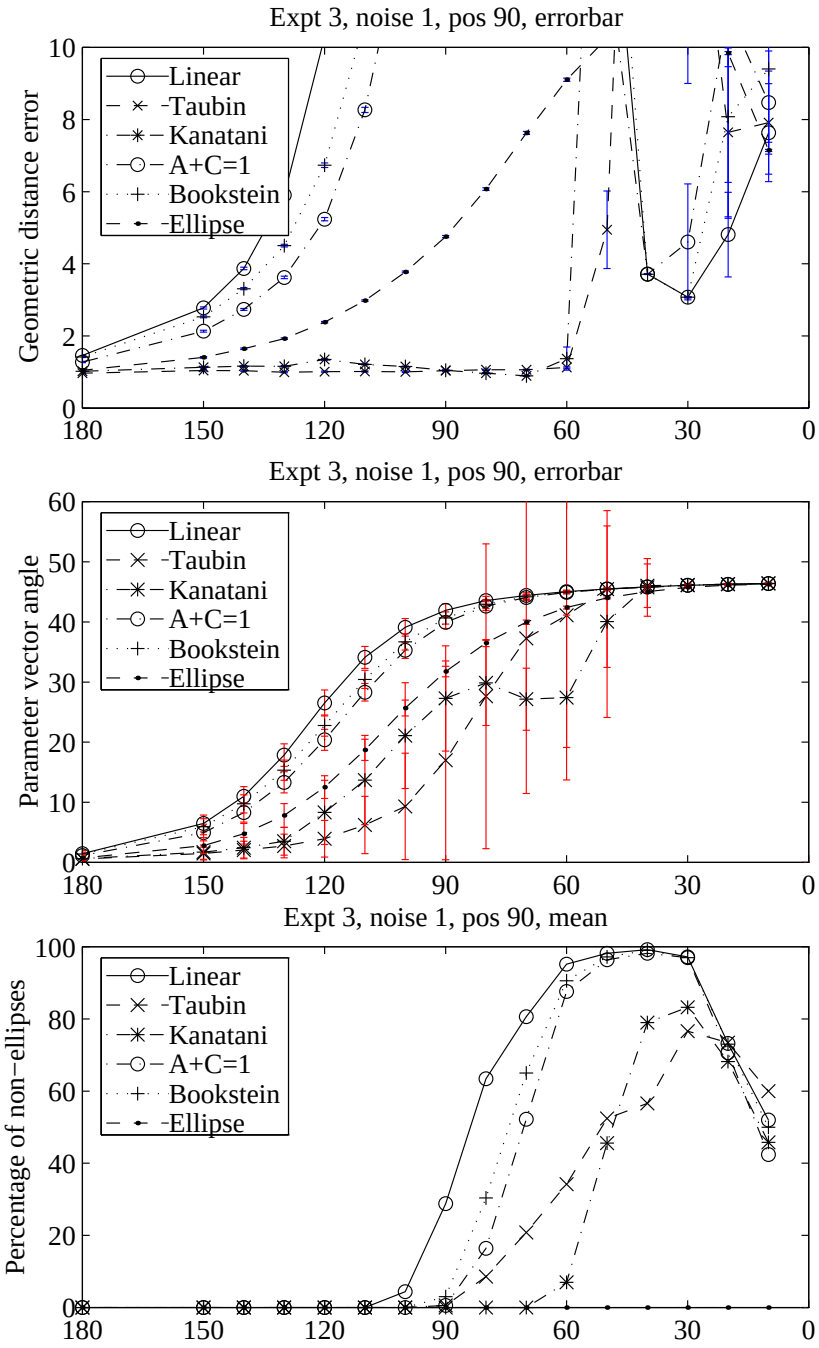


Figure 4.1: Comparison of different error metrics, low noise case. The abscissa may be read as amount of occlusion: the number of degrees of arc of the ellipse which were presented to the fitting algorithms. Top: Geometric distance between fitted conic and ground truth points. Middle: Angle between fitted and ground truth parameter vectors. Bottom: Percentage of non-ellipses returned.

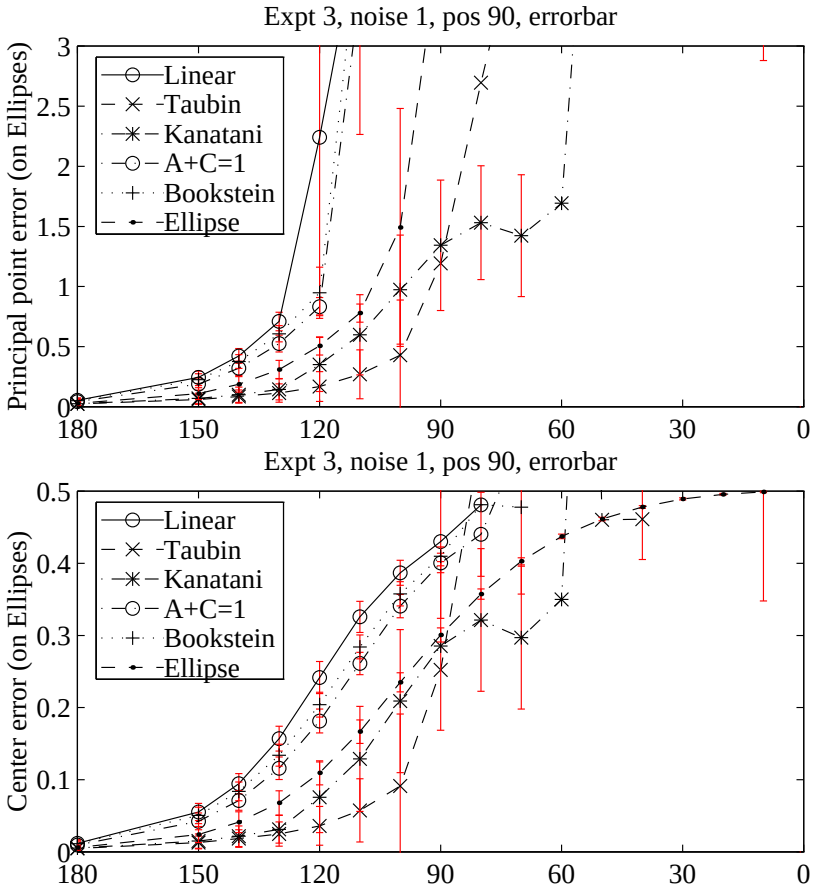


Figure 4.2: Comparison of different error metrics, low noise case. Top: Principal point error. Bottom: Center error.

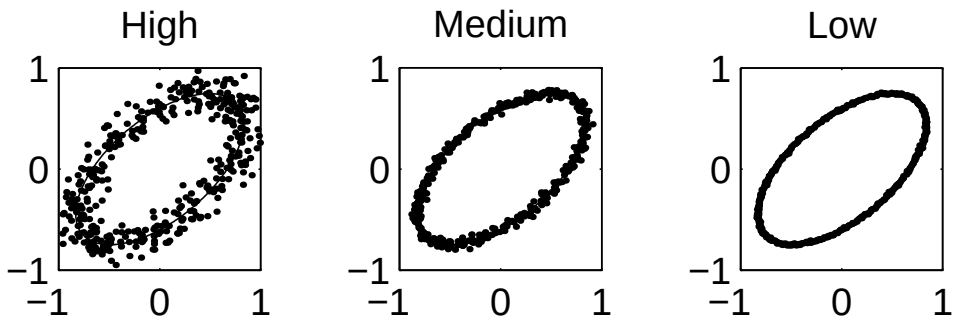


Figure 4.3: Visual depiction of the three noise levels used in the later experiments. The High, Medium and Low levels correspond to standard deviations of 4, 1 and $\frac{1}{4}$ pixels respectively.

Numerical conditioning

It has been observed [111, 50] that many computer vision algorithms suffer from numerical ill-conditioning simply because they compute in pixel coordinates. In the case of ellipse fitting, Rothwell notes that the matrix $\mathbf{S} = \mathbf{D}^\top\mathbf{D}$ contains terms raised to the fourth power, and with typical pixel coordinate values of the order of 100 to 1000 this leads to components of $\mathbf{S}$ of the order of 10^8–10^{12}. The condition number [43] of this matrix is then of this order, leading to computations which will lose all significant figures if solved in single precision floating point. Even in double precision, answers will be valid to only a few places. Ellipse fitting should, therefore, be performed only using *conditioned* coordinates, where the image points have been scaled to have approximately zero centroid and unit length. All computations in the experiments were performed in well-conditioned coordinates by considering only ellipses in the unit square.

This has the disadvantage that noise levels which are most easily comprehended in units of pixels are now in nonintuitive fractions of the ellipse diameter. To assist in the interpretation of the results in this section, I have arbitrarily scaled all noise levels by a factor of 40 to convert to pixel units. This implies that noise levels are relative to pixels on an ellipse with major axis 40 units—larger ellipses will correspond to smaller noise levels and *vice versa*.

Conic representation

For convenience in defining the experiments, central conic parameters are expressed as 5-vectors $[c_x, c_y, R_x, R_y, \theta]$, where (c_x, c_y) is the conic center, R_x and R_y are the X and Y radii respectively, and θ is the counterclockwise angle in degrees to the ellipse main axis from the positive X axis.

4.5.1 Experiment 1: Noise

In this experiment, we are interested in characterizing the behaviour of the algorithms with complete data and with respect to noise. Experimental procedure was as follows:

1. The ellipse $[0, 0, 1, \frac{1}{2}, 40]$ (see Figure 4.3) was sampled at 100 points uniformly spaced around the circumference.

2. Noise sigma was logarithmically varied between 2^{-4} and 2^4 pixels.

3. The sampled ellipse was corrupted with noise as described above for 500 runs and the distance between the true ellipse principal points and those of the conic returned by the fitting algorithm was recorded. Returned hyperbolae were ignored, averaging only over returned ellipses.

Figures 4.4 and 4.5 show the error in the principal point positions as a function of noise level. Figure 4.4 shows that at low to moderate noise levels ($\sigma < 5$ pixels), all the algorithms show an approximately log-linear relationship between noise level and principal point error. This error range is roughly the limit of what one might expect on real images. However, as discussed later, the "effective σ" will increase if smaller or more heavily occluded ellipses are imaged.

Figure 4.5 shows the same data as Figure 4.4, but the results have been scaled so that the ellipse-specific algorithm has a value of one. This more easily allows us to compare the algorithms. Again, under lower noise, all algorithms can be seen to perform similarly, while at high levels the $a + c = 1$ and Ellipse-specific algorithms retain their log-linear behaviour, while the others degrade less gracefully.

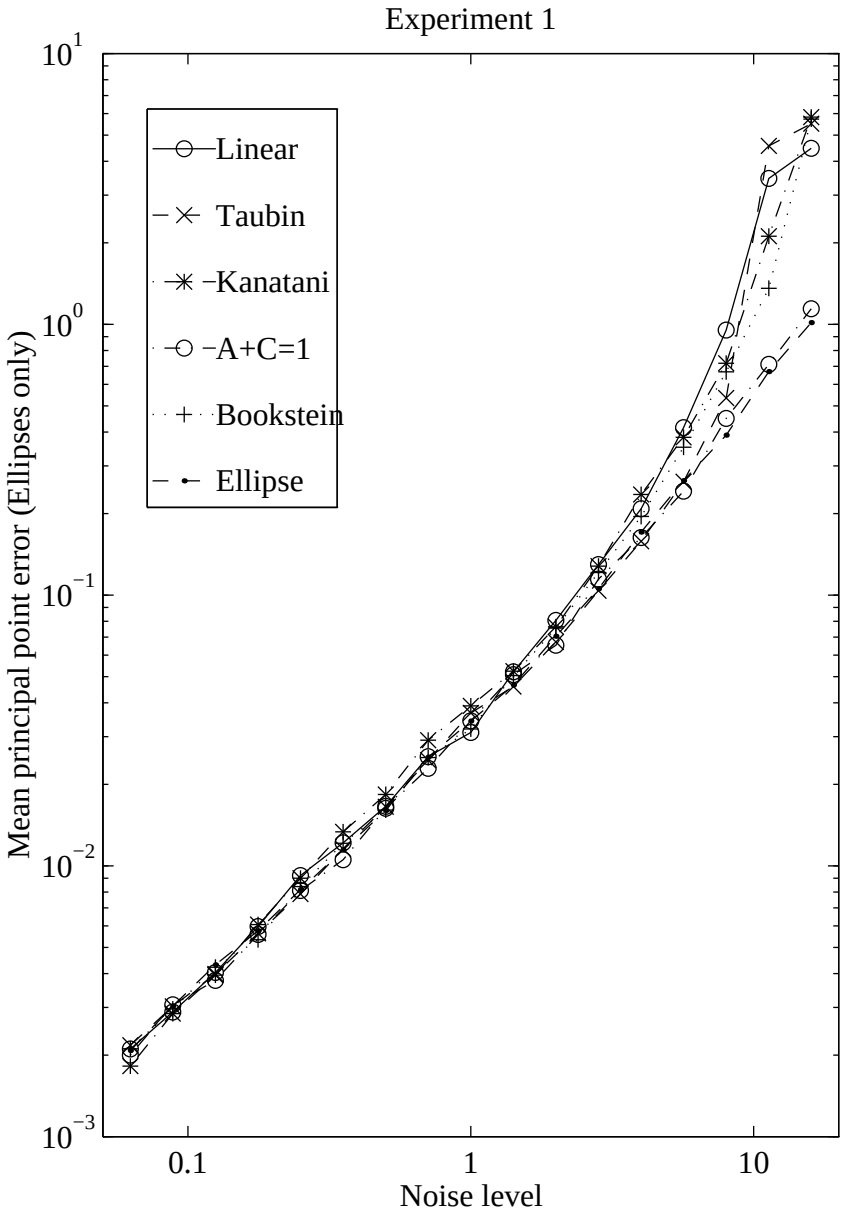


Figure 4.4: Raw results of experiment 1.

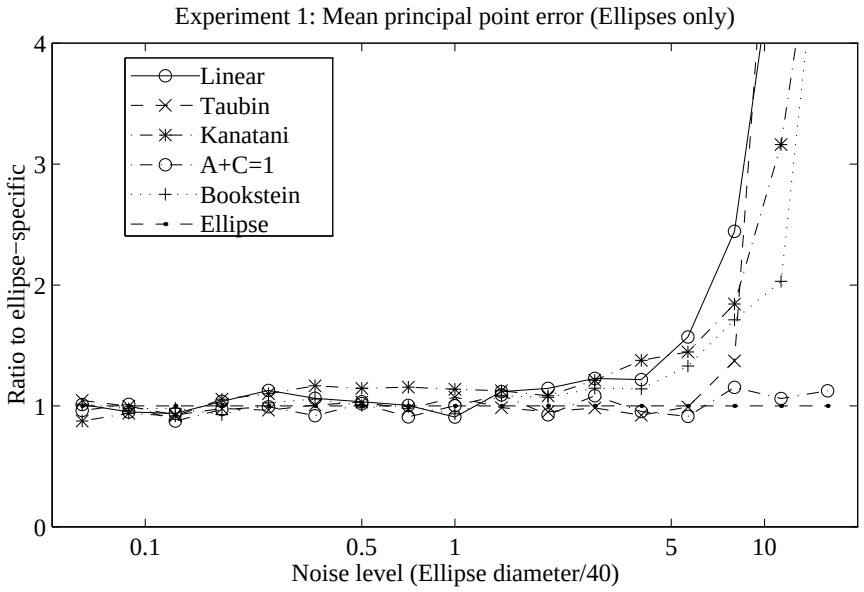


Figure 4.5: Results of experiment 1, relative to the ellipse-specific algorithm. At very high noise levels, the $a + c = 1$ and ellipse-specific algorithms are most stable. But at average levels of about 1 pixel, all algorithms fare similarly as noise increases.

4.5.2 Experiment 2: Orientation

In this experiment, we investigate how the errors in determining the center[1] of an elliptical arc vary as the portion of the ellipse from which the arc is sampled rotates about the ellipse. This is so that in Experiment 3 we may ensure that the subtended angle measurements are taken at the most pessimistic location about the ellipse. Experimental procedure is now described. For each of two values of the arc subtented angle, corresponding roughly to low and high occlusion:

1. The counterclockwise orientation of the center of the arc was varied from 0 to 180° in steps of 5°.
2. The ellipse $[0, 0, 2, 1, 0]$ was sampled at 100 points uniformly spaced along the arc.
3. The sampled arc was corrupted with the 'low' noise level as described above.
4. The ellipse center returned by the fitting algorithm was averaged over 100 runs. For the general conic algorithms runs were repeated until they returned ellipses, so that 100 ellipses were always used.

I show the results in two ways. Figures 4.6 and 4.7 illustrate visually the located center positions for the algorithms, while Figure 4.8 shows the error in center position as a function of the arc orientation.

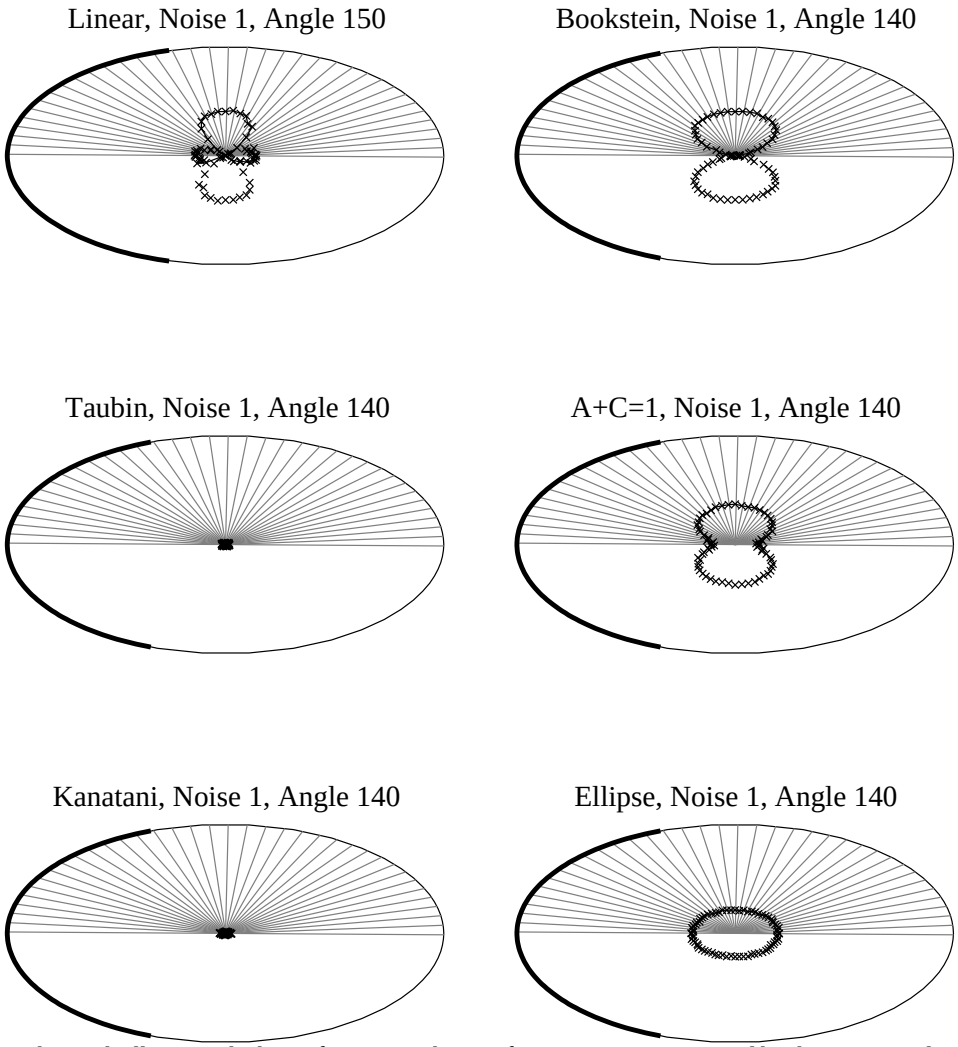
The visual depiction in Figures 4.6 and 4.7 highlights a number of interesting characteristics. The Taubin and Kanatani algorithms demonstrate no particular dependency on the sampling position, as indicated by the fact that their reported center positions are randomly scattered about the true center. On the other hand all the remaining algorithms have significant systematic errors (or *bias*), and this bias is a relatively smooth function of the position of the arc along the ellipse. Perhaps more surprisingly, for the biased algorithms the returned center is always on the line joining the arc midpoint to the true conic center. This in turn points to the reasons for the effectiveness of the Kanatani bias-correcting algorithm—if the bias is a smooth function of the parameters, then it will be consistently inferred, even from the poor initial estimate with which the algorithm is initialized.

In Figure 4.8, the distance error is plotted as a function of the arc angle, allowing the magnitude of the error to be seen. Again we confirm that the Taubin and Kanatani algorithms are effectively insensitive to the arc position. However, the results for the other fitters are somewhat surprising. Intuition would suggest that the worst place to sample from is the minimum curvature region at the top (and bottom) of the ellipse (where the angle is 90°), and indeed for the linear algorithm this is the case. The Bookstein and $a + c = 1$ algorithms, though, have the greatest error at the 45° and 135° quarter points; while the ellipse-specific algorithm in fact performs best on the low-curvature region.

Examining this last result, it is apparent that the error magnitude is reduced at the low-curvature regions simply because the ellipse data points are then closer to the center. Scaling the error at each angle by the radius at that position gives the graphs in Figure 4.9, which agree more readily with the intuitive expectation that the reduced "information content" at the low-curvature regions will lead to poorer results. Now the Linear, Bookstein and $a + c = 1$ algorithms all exhibit their worst relative errors on the low-curvature region. The ellipse-specific algorithm then has an error that is constant over all choices of angle.

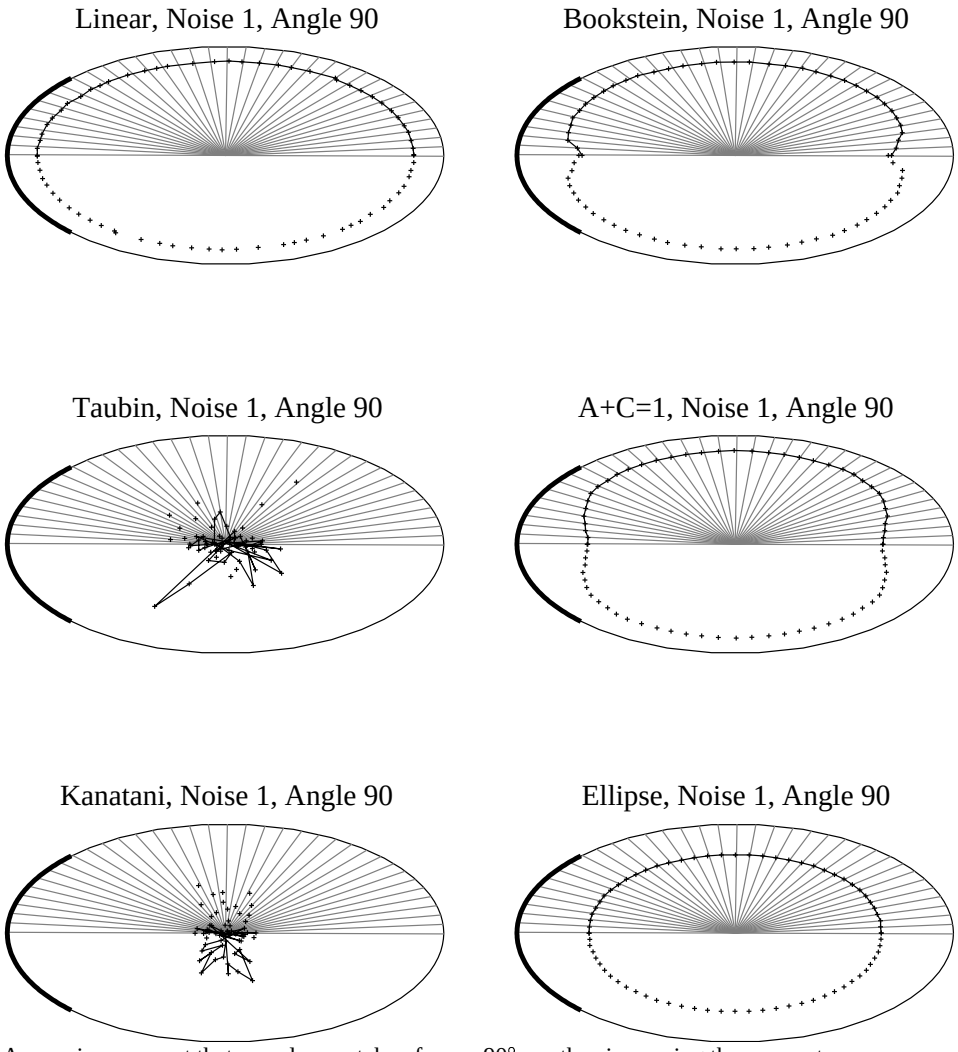
The conclusion is that for the algorithms which indicate a dependency, maximum errors occur when the arc is sampled from the low curvature sections at 90° and 270°. For the other algorithms, any choice of angle is equivalent. Therefore, in subsequent experiments we sample from these points in order to ensure worst-case behaviour.

[1] In this section, the center position is used because it is easier to visualize than the principal points.



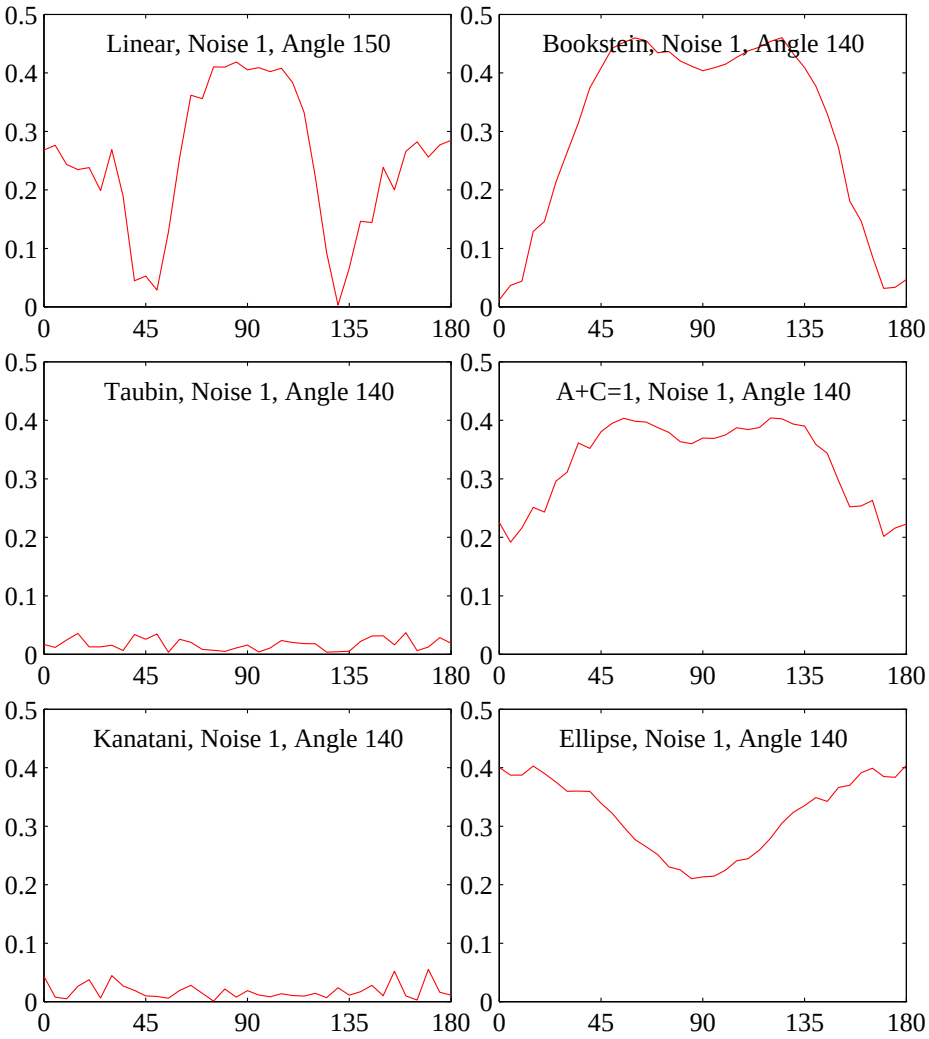
Within each ellipse, each chain of crosses is the set of center positions returned by the corresponding algorithm. Each returned center point (i.e. each $\times$) corresponds to a 140 degree arc of the ellipse (except for the Linear algorithm for which 140° is beyond its breakdown point and 150 was used). The actual arc involved is generally that which is bisected by the line from the true centre through the fitted centre, and on to the ellipse (because all algorithms tend to have some high-curvature bias). The thick curve simply gives an impression of the amount of arc used, and is in no special position.

Figure 4.6: Results of experiment 2, low occlusion.



As previous, except that samples are taken from a 90° arc, thus increasing the error rate.

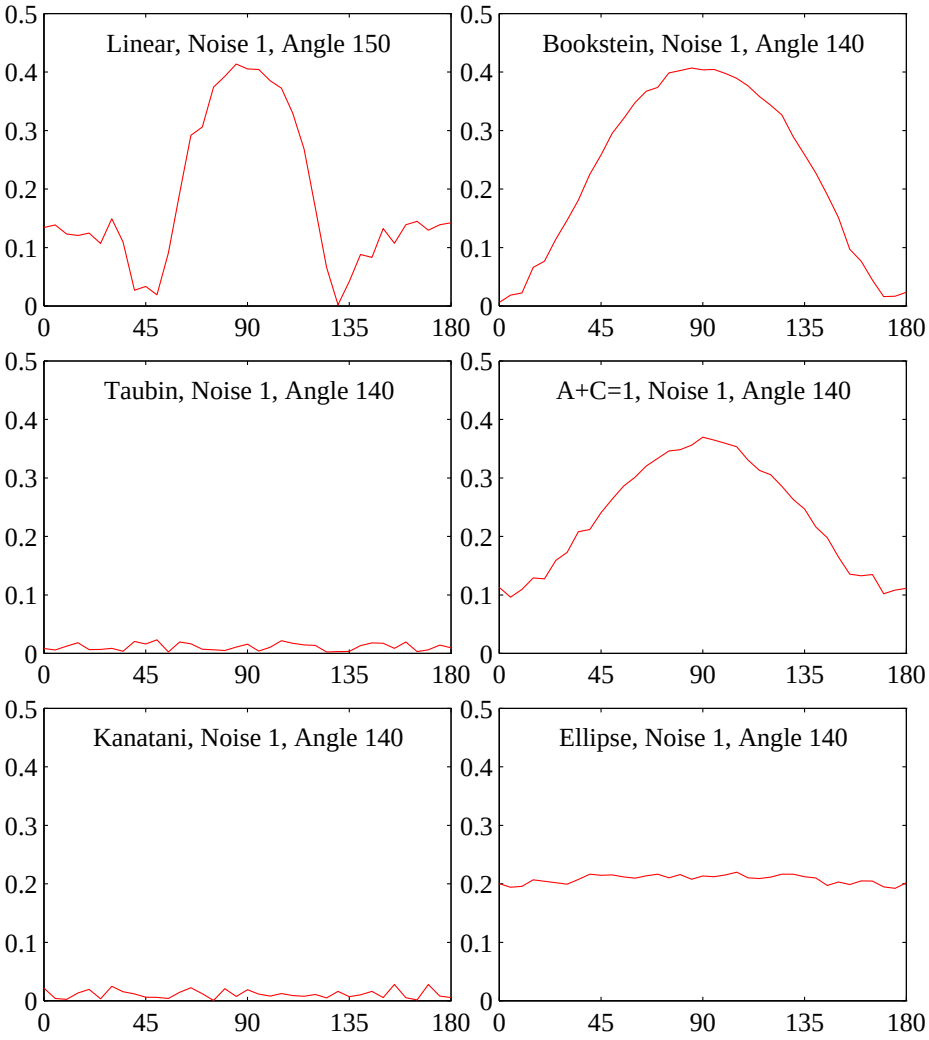
Figure 4.7: Results of experiment 2, high occlusion.



The metric is the distance of the center from the true center (which is at the origin). The Taubin and Kanatani algorithms are basically insensitive to the position of the arc, producing randomly varying results.

For the ellipse-specific fitter, the worst case is in the high curvature region, due to its consistent high-curvature bias, which pulls the centre towards it. For the other conic fitters, the worst place to sample from, in terms of center error, is the low-curvature region.

Figure 4.8: Results of experiment 2. Center error versus arc position.



As above, with the error normalized by the radius at the point on the arc from which the sample is taken. We observe that the ellipse-specific fitter now has a roughly constant error—implying that its error is proportional to the radius.

Figure 4.9: Results of experiment 2 normalized by radius.

4.5.3 Experiment 3: Occlusion

The third experiment is designed to locate the breakdown point of each of the algorithms when the ellipse is progressively occluded. We measure the errors in center position and radius estimates for several arcs of decreasing subtended angle. Experimental procedure was as follows:

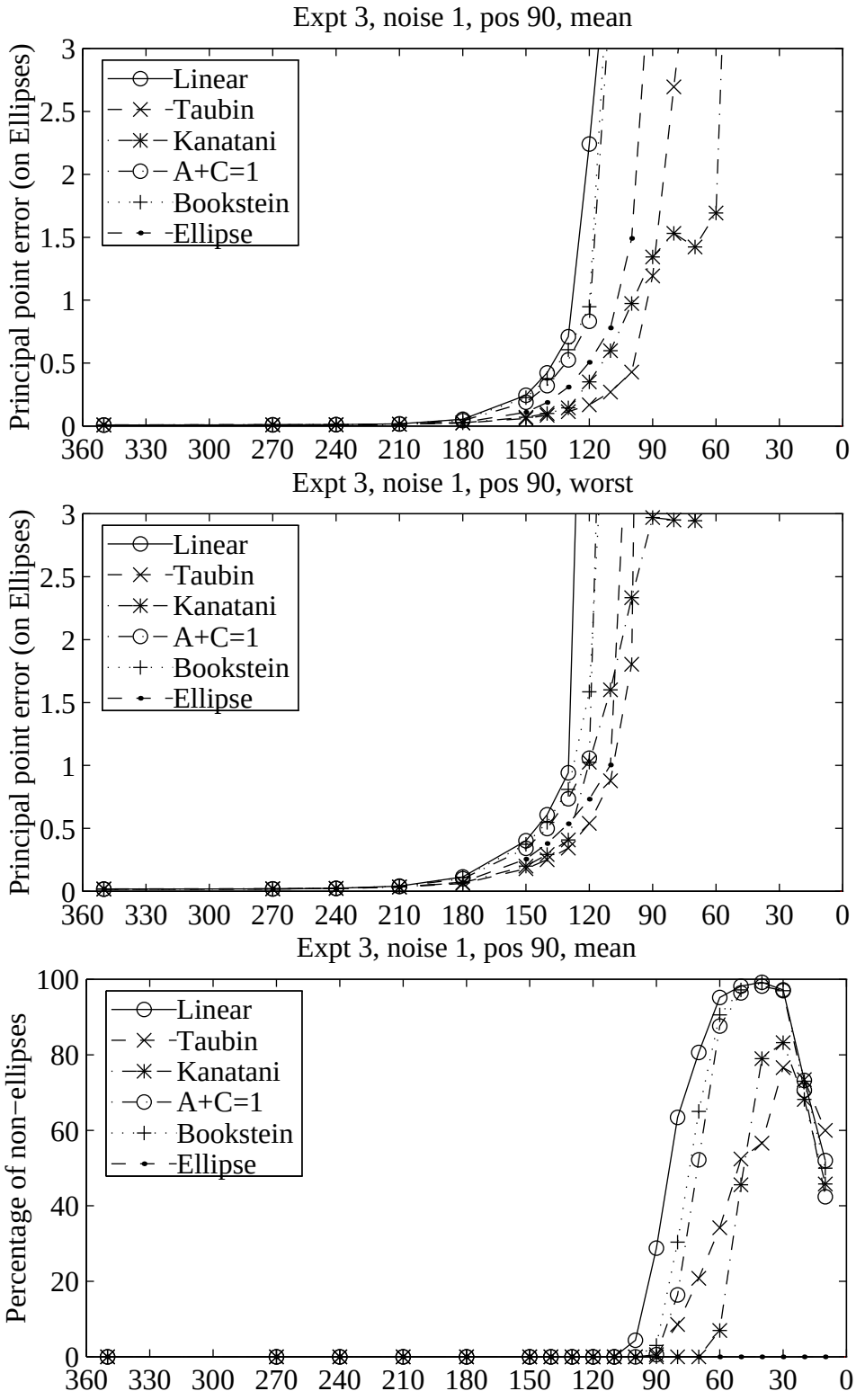
1. The angle subtended by the elliptical arc was varied from 360° down to 0 in steps of 10°.
2. The ellipse $[0, 0, 1, \frac{1}{2}, 40]$ was sampled at 100 points uniformly spaced along the arc.
3. The sampled arc was corrupted with the three 'standard' noise levels as described above.
4. Over 1000 runs, the noisy arcs were submitted to each fitting algorithm. The distances of the fitted principal points to their correspondents on the true ellipse were calculated. The mean sum distance where the algorithms returned ellipses was calculated, as was the percentage of runs in which non-ellipse conics were returned.

Figures 4.10,4.11,4.12 show the plots of principal point error and percentage of returned non-ellipses as a function of decreasing subtended angle, for the three noise levels. In addition, Figure 4.13 shows the parameter angle error for all three levels. The graphs show that all the algorithms break down as the amount of occlusion increases, but that in low noise, the Kanatani and Taubin algorithms show the best performance, breaking down later than the others. As noise increases, the increased stability of the ellipse-specific algorithm means that it also becomes competitive with these algorithms in terms of accuracy.

As occlusion is decreased beyond the breakpoint, the algorithms show some interesting behaviour, illustrating their responses on essentially random data. In this case we can see the bias towards or away from ellipse fitting of each algorithm. For example, in Figure 4.12, the $a + c = 1$ constraint returns ellipses only, indicating the bias towards ellipse-specificity noted by other authors, while the linear algorithm shows a strong bias towards hyperbolae. The Kanatani algorithm returns about 50% of each type of curve which is consistent with its unbiasedness.

The graphs for worst-case performance, on the other hand, begin to favour the ellipse-specific algorithm much earlier. Although the Taubin and Kanatani algorithms are the most accurate on average, the wide variance in their responses means that the 3σ errorbars overlap with the ellipse-specific method even on low noise.

Finally, the results for the more expensive "Sampson" and geometric distance algorithms are shown in Figure 4.14. The measure used is the geometric distance, as this is the measure which both algorithms attempt to minimize, and therefore it is instructive to note the extent to which the more tractable algorithms achieve this goal. The result is that neither improve greatly on the Taubin algorithm, and the Sampson algorithm (initialized using Bookstein) fails to reach even the level attained by Taubin. We may conclude then that the more expensive algorithms do not in fact offer any great advantage over the faster methods in terms of minimizing the geometric distance. However, when combined with robust fitting, which necessitates an iterative algorithm, they may again become competitive.



The upper curve shows the average error. The middle curve shows the "worst-case" performance: three standard deviations above the mean performance. The lower curve shows the percentage of non-ellipses on elliptical data.
We note that the Taubin and Kanatani algorithms offer superior mean accuracy, along with a later breakdown point. Note that after the error reaches about 1.5, the algorithms are essentially performing randomly.

Figure 4.10: Results of experiment 3, low noise case.

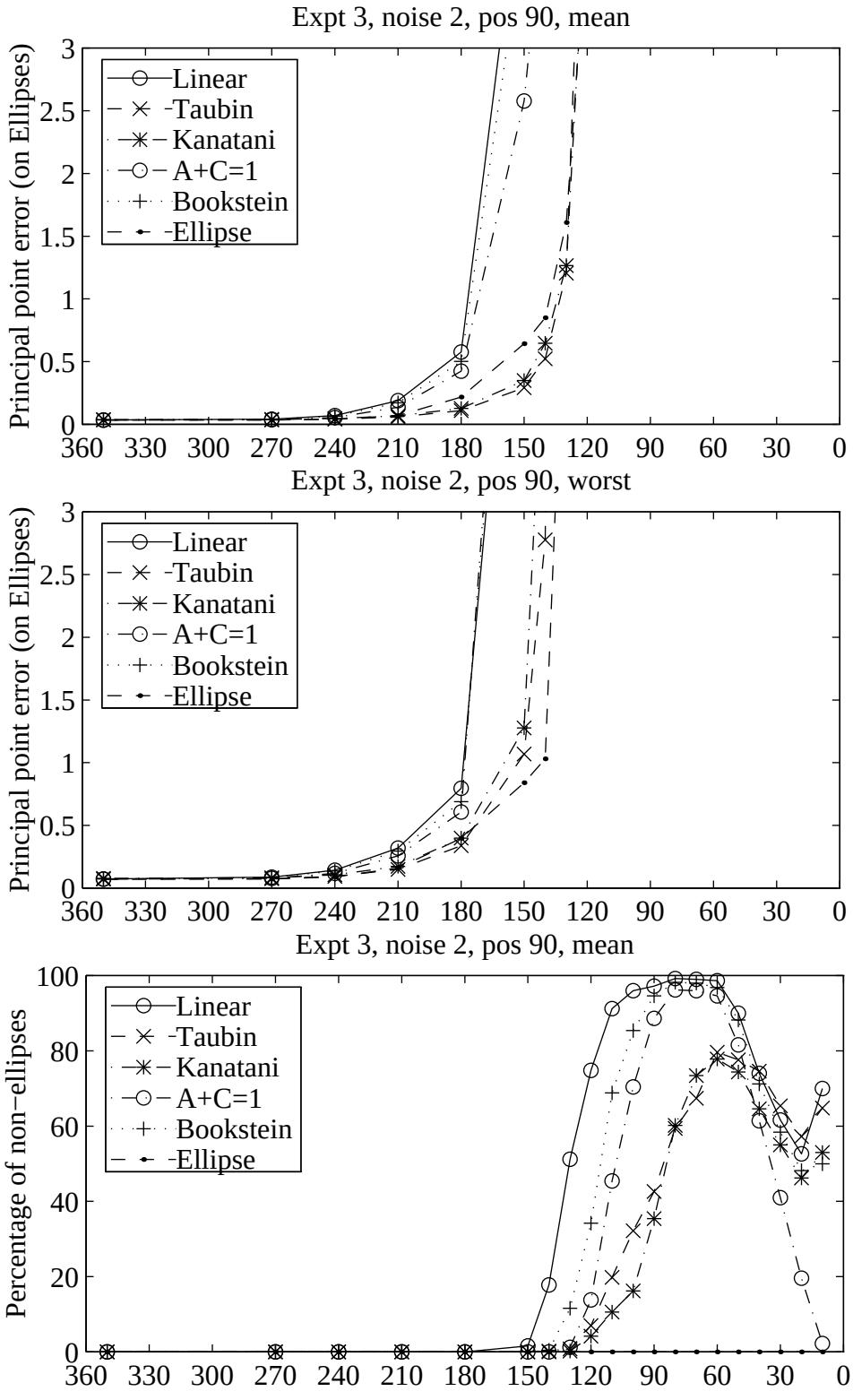


Figure 4.11: Results of experiment 3, medium noise case. Curves are as above.

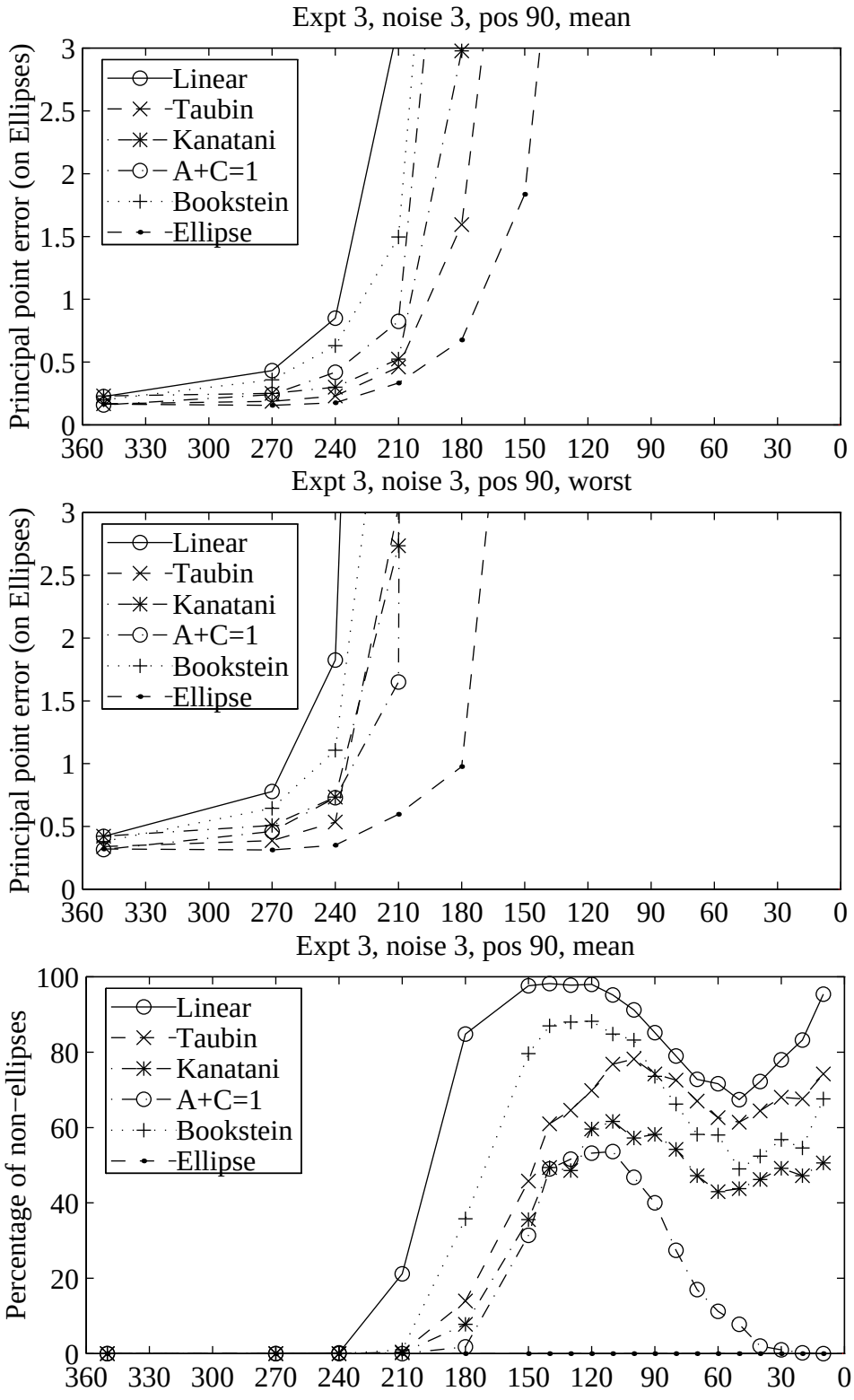


Figure 4.12: Results of experiment 3, high noise case. Curves are as above.

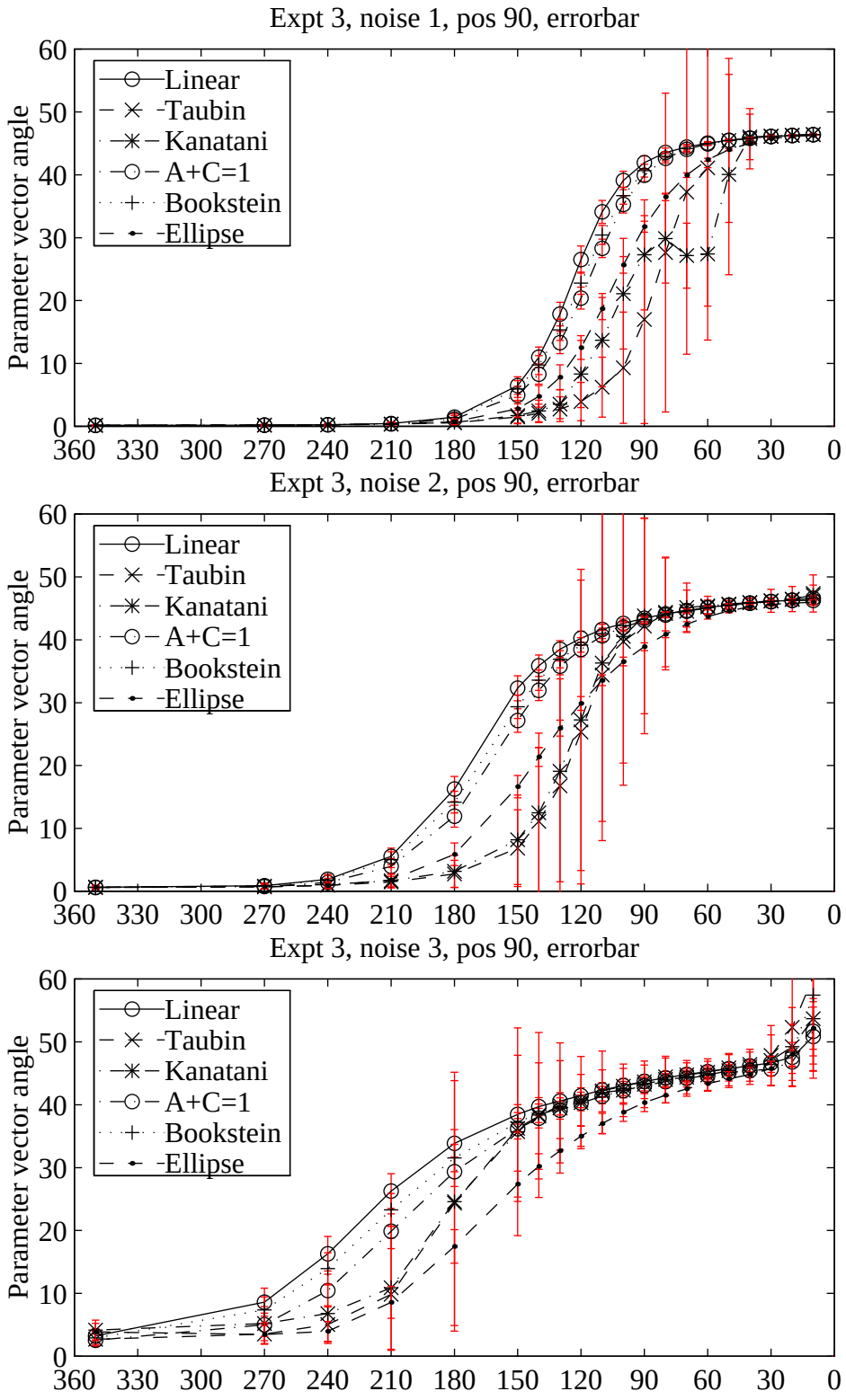


Figure 4.13: Results of experiment 3, parameter angles. The curves show parameter angle error from the three noise levels, indicating general conic fitting performance.

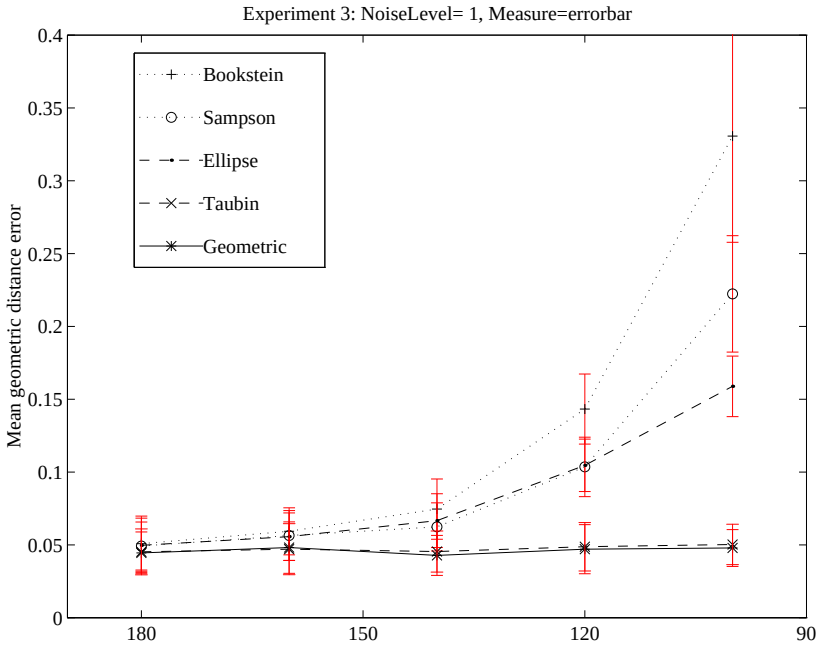


Figure 4.14: Results of experiment 3, expensive algorithms. This graph shows results for the more expensive "Sampson" and "Geometric" algorithms for the smallest noise level and only in the breakdown region. We note that even at low noise and occlusion, Sampson is beaten by the cheaper Taubin algorithm. The "Geometric" algorithm improves slightly on the Taubin algorithm, but not by a great deal.

4.5.4 Experiment 4: Real data

In order to provide examples of the algorithms' performance on real data, a real ellipse image has been processed by a local implementation of the Canny edge detector [21]. Edgel chains corresponding to the boundaries of the two ellipses have been manually extracted. In order to simulate occlusion, we adopt the following experimental strategy for each ellipse.

First, fit an ellipse to the entire chain using any one of the fitters (the results are indistinguishable with different fitters). Use this ellipse to determin the principal point. For values of the x coordinate between the minimum and maximum in the chain:

1. Extract the edge points for which $x_i < x$.
2. Draw the fitted conic on the original image.
3. Record the distance between the recovered principal point and that obtained from the full fit.

The resulting image (Figure 4.15) gives a visual depiction of the algorithm results as the ellipse is progressively occluded. The quantitative performance graphs are shown in (Figure 4.16). Again we note that the Taubin algorithm provides a more accurate solution, with reduced bias over the other fitters. We also confirm that the Ellipse-specific algorithm is most consistent, producing better estimates than Taubin at the highest occlusion levels. However, the Kanatani algorithm appears to be more biased than the synthetic results would indicate. This may be because the quantization error in the shorter edgel chains defeats its noise model.

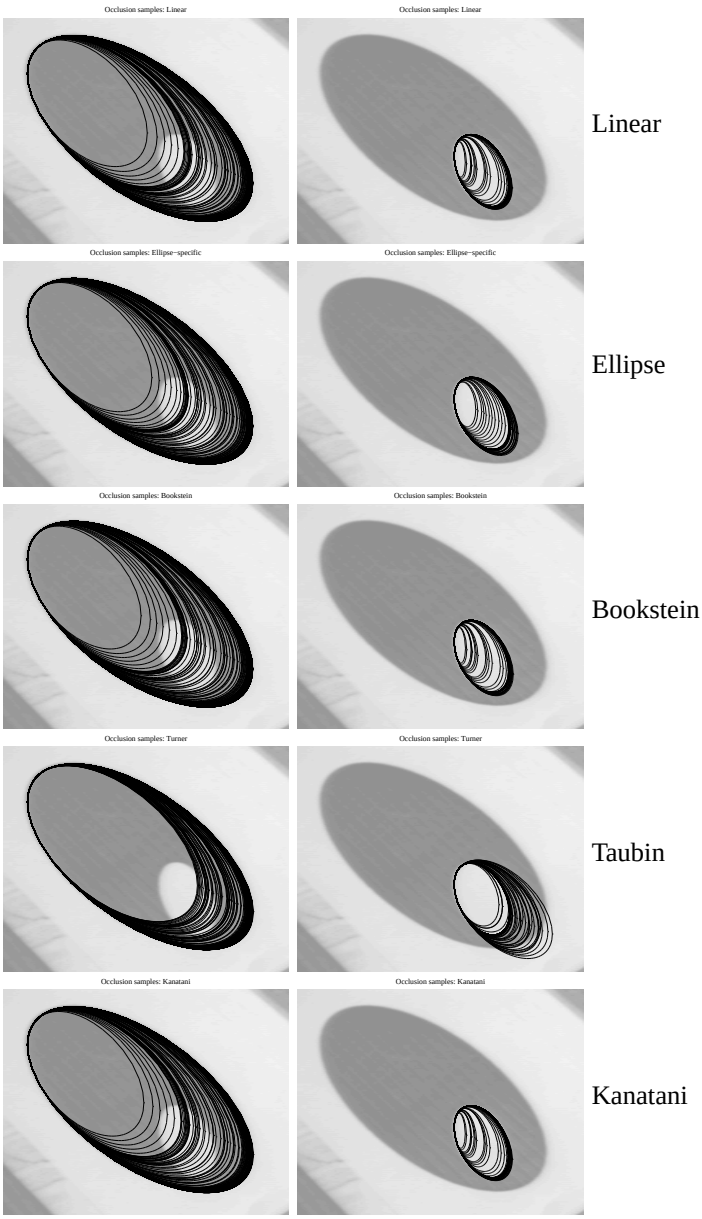


Figure 4.15: Occlusion tests on real data, qualitative. Edges of the two ellipses are extracted using a local implementation of the Canny edge detector, and occlusion is simulated by removing portions of the edgel chain to the right of a vertical line. The fitted edge is then overlaid on the image. We note again the improved bias on the latter two algorithms and the stability of the Ellipse-specific algorithm.

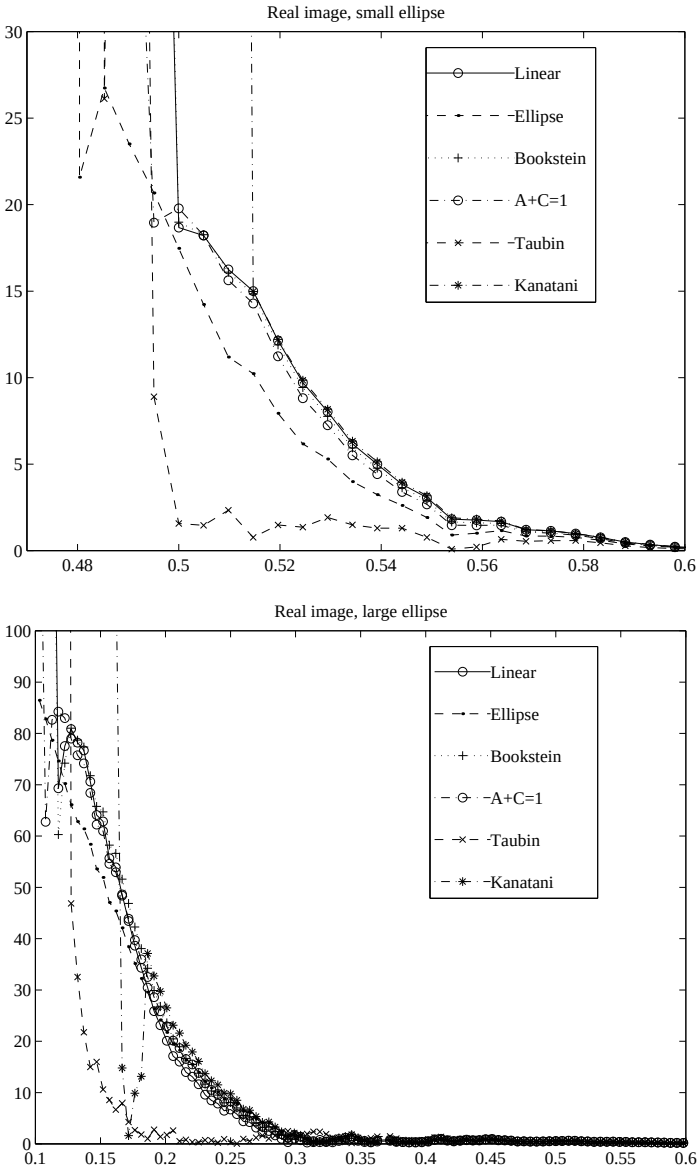


Figure 4.16: Occlusion tests on real data, quantitative. Distance of recovered center from true center for each algorithm on each of the two ellipses. The upper graph shows performance on the small ellipse, with Taubin best, and the Ellipse-specific second. The lower shows that on the large ellipse, Taubin is still best, but there is little to choose between the other algorithms.

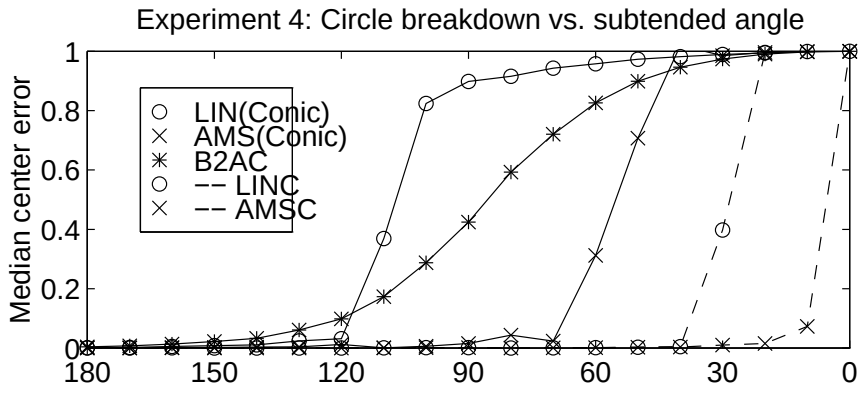


Figure 4.17: Results of experiment 4. The specialized circle fitters LINC and AMSC ("Specialised Taubin") have better breakdown characteristics than the corresponding general conic algorithms. (B2AC is the Ellipse-specific algorithm, and AMS is the Taubin algorithm).

4.5.5 Experiment 5: Circles

In the final experiment, we consider the breakdown performance when the general conic fitting algorithms are made specific to a particular task; in this case, circle fitting. The procedure is similar to Experiment 3, but we add two new algorithms which are specializations of the Linear and Taubin algorithms respectively to circles. This specialization is trivially achieved by replacing the least squares design matrix $\mathbf{D}$ with the following:

$$\mathbf{D} = \begin{pmatrix} x_1^2 + y_1^2 & x_1 & y_1 & 1 \\ x_2^2 + y_2^2 & x_2 & y_2 & 1 \\ & & \cdots & \\ x_n^2 + y_n^2 & x_n & y_n & 1 \end{pmatrix}$$

And replacing the derivative matrix of the Taubin algorithm appropriately. Figure 4.17 shows the breakdown curves for the low-noise case. As expected, the specialized fitters break down considerably later than the general conic algorithms, illustrating once more the improvement in stability provided by reducing the scope of the shape representation.

4.6 Discussion

This chapter has discussed the problem of fitting conic sections to ellipse data. The experiments illustrate that the key parameter affecting the algorithms' accuracy is the amount of occlusion present and the qualitative noise level. With complete data, all algorithms exhibit a similar degradation in the presence of increasing noise.

As the data become progressively incomplete, a breakdown point is reached beyond which the algorithms fail catastrophically. This breakdown point is superior with the Ellipse-specific, Taubin and Kanatani algorithms, and, in the special case of a circle, with the circle-specific algorithms. Under high noise, Kanatani has superior accuracy but returns hyperbolae up to 60% of the time when presented with highly occluded ellipses.

Algorithm complexities are, in increasing order: $a + c = 1$, Bookstein, Linear, Ellipse-specific, Taubin, Kanatani, Sampson, Geometric distance. We note however that the Kanatani algorithm is almost an order of magnitude slower than the preceding five, and that the Sampson and geometric distance algorithms are significantly slower than Kanatani.

Taubin's algorithm is the best tradeoff between speed and accuracy in almost all cases. It is almost as accurate as Kanatani's but is about 10 times faster, and is only twice as slow as the fastest algorithms. In cases where high accuracy is required, Kanatani is the method of choice. Where stability is of importance, the Ellipse-specific algorithm is a clear leader. In order to achieve both accuracy and robustness, the recommendation of this research can be summarized as: use Taubin, and when it fails use the Ellipse-specific algorithm.

4.6.1 Future Work

This chapter has concentrated largely on the conic fitting algorithms which provide closed-form or well understood solutions to the fitting problem. However, there is another family of conic fitters—based on iterative minimization of a distance function—that might provide superior results. This was not done here because the performance of the Kanatani and Taubin algorithms was such that more expensive techniques were felt to yield little improvement in accuracy over that provided by the latter two. Also, there exist good theoretical surveys of the distance-metric based algorithms [104, 105]. However, it would be useful to incorporate into this experimental framework the nonlinear techniques, particularly the ellipse-specific algorithms of [103, 40, 115], along with Porrill's [91] alternative bias correction algorithm.

The range of error metrics used, while wide, does not necessarily indicate the suitability of the fitters for all tasks. Examination of some alternative error metrics such as the conic invariants [96, 94] might allow us to widen the range of applicability of these results.

Chapter 5

The Run-Distribution Test

This chapter is concerned with the problem of deciding whether a fitted model accurately describes the data to which it has been fitted. We have developed an effective method of testing the lack-of-fit of a parametric model to data, with applications to the computer vision problems of robust estimation, model selection, and curve and surface segmentation.

The benefits of this technique are high sensitivity (large response to small outliers) and very low dependence on the noise distribution of the input data. Our test is new to the computer vision community in several ways:

- We look at the *distribution* of the residual errors, rather than basing statistics directly on their *values*.
- We assume a broad enough class of distributions as to be essentially distribution independent.
- The test requires *no knowledge of the sensor noise level*, and its response is essentially independent of that level.

We present results of experiments that compare the test with the standard chi-squared (χ^2) statistic, and the median absolute deviation (MAD) measure used in robust estimation. The experiments are designed to represent typical vision tasks, namely feature tracking, robust fitting, and segmentation. We show that our test is comparable to the MAD and chi-square, but is cheaper than the MAD, and requires no knowledge of the noise level, unlike the chi-squared. These show that the new test is superior in cases where the noise level is poorly known, but that if it is known, more accurate results will be obtained using a test such as the chi-squared.

5.1 Introduction

It is very common in computer vision to wish to represent some large dataset in a concise way in order to extract geometric properties, attenuate noise, or simply to reduce the volume of data. In almost all cases, this is achieved by fitting an appropriate parametric model to the data set in the least squares sense. It is then vital to have some way of telling when the fit is wrong, and the model is not 'appropriate' to the data. Simple least squares techniques [93] assume the noise in the data to be strictly Gaussian of known variance, and then use the χ^2 test to give an estimate of the probability that, under that assumption, the data fits the model. Robust estimators [76] approach the problem more directly, by effectively ignoring data points which do not fit the model. Robust models are, however, even more expensive to fit than unbiased nonlinear models, and do not help when the model is already fitted to the data, and simple verification is all that is needed. My argument asserts that least squares is adequate for most purposes, *until its assumptions are violated*. Of course it is precisely

these boundaries, at which the assumptions are violated, that are of most importance to the visual process. Hence, a quick and effective test which identifies such errors will allow a cheap estimator to be used on most of the signal, while the more expensive techniques are held in reserve until the cheaper methods fail.

5.2 Goodness-of-fit Testing

As in Chapter 2, we denote the data points to which the model is to be fitted by $\{\mathbf{x}_i\}_{i=1}^n$ and the parameters of the model by $\mathbf{a} = \{a_i\}_{i=1}^p$. We also assume that we have a distance metric $D(\mathbf{a}, \mathbf{x})$ which measures the signed distance between a particular data point and the fitted model. The model fitting process is assumed to have found the value of $\mathbf{a}$ for which the error metric

$$\epsilon = \sum_{i=1}^{n} \phi(D(\mathbf{a}, \mathbf{x}_i))$$

is minimized. The function $\phi(x)$ is an influence function, which for classical least squares is $\phi(x) = x^2$. We do not need to know the form of ϕ, simply that it must be symmetric or antisymmetric about $x = 0$. Having found the value $\mathbf{a}$, we can define the set of *residuals*

$$R = D(\mathbf{a}, \mathbf{x}_i)_{i=1}^n$$

The task of goodness-of-fit testing is to determine, based on the values of the residuals, whether it is likely that the model describes the data. Lack-of-fit statistics say whether the model is unlikely to describe the data[1].

5.2.1 Chi-Square Test

Whaite [131] provides an accessible summary of the chi-square testing technique. The basic assumption is that each observed point $\hat{\mathbf{x}}_i$ is the exact point corrupted by an isotropic zero-mean Gaussian noise process of variance σ^2. This means that if σ is known, the random variables R_i/σ are zero-mean Gaussian random variables with unit variance. Then the chi-square statistic $\chi^2 = \sum_{i=1}^n (R_i/\sigma_i)^2$ has a known distribution. In fact the number

$$Q(\frac{n-p}{2}, \frac{\chi^2}{2})$$

where Q is the increasing incomplete gamma function $Q(a, x) = \frac{1}{\Gamma(a)} \int_0^x t^{a-1} e^{-t} \mathrm{d}t$ [93] gives a measure of how badly the model fits the data.

The disadvantages of the χ^2 test are well known: the Gaussian noise model has repeatedly proved unrealistic in computer vision and the noise variance is often difficult to know in general. Additionally, the test, depending on a linearization of the residual equation, fails in the presence of high noise, even when the noise level is known (see Figure 5.7).

5.2.2 Median Absolute Deviation

The median absolute deviation (MAD) measure is not strictly a test, in the sense of providing a probability of error. However, because it is essentially the error metric used in robust estimators, it is interesting to see how its response compares with the test proposed here. The measure is simply the median of the absolute values of the residuals, and may be evaluated in $O(n)$ time [7]. To use this measure as a test of goodness of fit, we need an

[1] The distinction between lack of fit and goodness of fit is subtle and of great interest to statisticians, but we shall not make it here, treating the two terms as equivalent.

estimate of the noise level. For Gaussian distributed residuals with a standard deviation σ, the median M of the absolute values of the residuals satisfies

$$\frac{1}{\sigma\sqrt{2\pi}} \int_0^M e^{-\frac{t^2}{2\sigma^2}} dt = \frac{1}{4}$$

From this, we can calculate the expected value

$$\begin{aligned} M &= \text{erf}^{-1}(\frac{1}{2})\sigma \\ &\approx \tfrac{2}{3}\,\sigma \end{aligned}$$

and threshold the MAD value accordingly.

5.2.3 Maximum Run Length Test

The "RANSAC" system of [14] and the 3D segmentation algorithm of Besl [8] use the most similar tests (to the test presented here) reported in the vision literature. Both, however, only consider the *maximum* run length (see below) observed for a set of residuals. In our experiments, we have found this measure to be noise sensitive. Also in both of these systems, the runs test is used only as a "backup" test in conjunction with the chi-squared test.

5.3 Run-distribution Test

We now introduce our test, which we have called the run-distribution test. We describe the idea behind the test, the noise model which we assume, the actual test, and how it differs from similar tests in the literature.

The tests discussed above essentially *first* extract one number from the set of residuals, and *then* use that as a basis for discrimination. Instead we want to look at the entire set of residuals R, and decide whether that set is what we would expect, given data which is in concordance with both our parametric and noise models. At that point, we can return a single measure, which describes the likelihood that the particular set of residuals is indeed consistent with having been drawn from the population of possible sets.

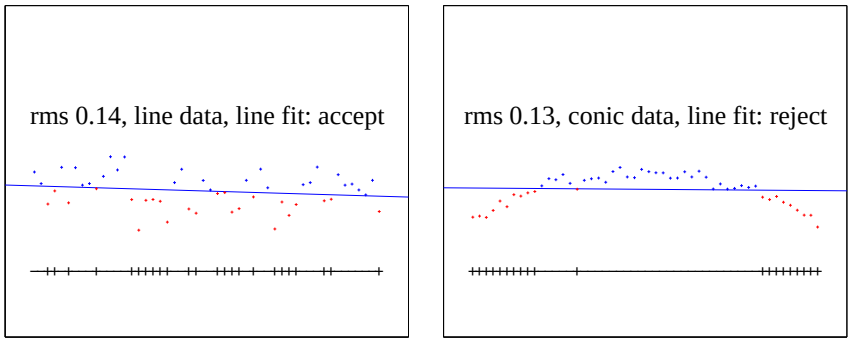


Figure 5.1: **Motivation for the RD test**. The first dataset is consistent with interpretation as a line, while the second is not—being more readily seen as a circle. This is despite the circle data having a smaller root-mean-square error than the line. Underneath, the $+$ and $-$ symbols show the residual signs used in the RD test.

Figure 5.1 shows the motivation for the test. Even though the line dataset has a larger fitting residual, it appears to be better explained by a line model than does the circle data. The data is well explained in the first case because the residuals after fitting look like random noise, but in the circle case there is still "structure" in the residuals. However, structure in the residuals contradicts the assumption that the R_i are drawn from a random process, so we can therefore say, without *a priori* knowledge of the noise level, that the first fit is good and the second is bad. The runs test is one simple method of looking for structure which will now be developed.

5.3.1 Noise model

We allow each point to be corrupted in each dimension by a scalar noise component sampled from a symmetric zero-mean process plus an outlier process. Note that this is a very wide range of distributions, trivially including the normal distribution. Moreover, this particular type of distribution is common in computer vision. With such a distribution, the residuals after least-squares fitting will be similarly distributed. We can therefore detect outliers by quantifying the extent to which the *distribution* of the residuals matches our noise model.

5.3.2 Definition

We begin by noting that if the noise distribution is zero-mean, the residual set R should have equal numbers of positive and negative elements—points should fall randomly on either side of the curve. By deleting any zeroes from R, we can make a set of the signs of residuals

$$S = \{\text{sign}(R_i)\}_{i=1}^{n}$$

whose elements may be represented as either + or -. Following von Mises [130, page 184] we define a *run* as a sequence of one or more residuals of the same sign. For example the set $S = \{$+-+++--+-$\}$ contains runs of lengths 1,1,3,2,1,1 respectively. Intuitively, we would expect that if the model fits well, there will be a large number of short runs, with many long runs of positive or negative residuals indicating that the model has been biased. The *distribution* of runs is a histogram of the number of runs observed for each run length. In the previous example, there are four runs of length 1 and one each of lengths 2 and 3.

Measuring the likelihood of a particular distribution of runs is a problem that has been approached in the statistical literature [19, 59, 78]. In particular, having decided to measure the runs, the question arises as to how to quantify the deviation of a particular example from the general population. Kempthorne *et al* [59, page 234] calculate the expected value and variance of the total number of runs M

$$\begin{aligned} E[M] &= n + 1 \\ E[M^2] &= \frac{n(n-1)}{2n-1}, \end{aligned}$$

and approximate the distribution by a Gaussian in order to calculate probabilities. This approach, taken also by [78], [19], and [130] simplifies the analysis, but reduces the sensitivity of the test. In this paper, we instead compare the "actual" distribution to the observed distributions.

5.3.3 Comparing the distributions

If we make a histogram $H(j)$ where bin j contains the number of runs of length j in the residuals then this will approximate the probability density function (*pdf*) of the true run distribution. The runs histogram for the example above is then $H = \{4, 1, 1, 0, \ldots\}$. By comparing this distribution with the pdf $P(j)$ produced by a random process, we can

determine whether the residuals are truly random, or still contain structure, implying that the original data are not well described by the fitted model. In deciding how to compare the histograms, the accepted choice is, again, to use the chi-squared test, but this time as a method of comparing distributions. From [93, §14.3], the statistic that we wish to test is the sum

$$\sum_{j=1}^{n} \frac{(H_j - P_j)^2}{P_j}$$

which is known to be distributed as a chi-squared variable with n degrees of freedom. Then using the chi-squared test as above, we obtain the probability that a particular histogram of runs H is consistent with the true histogram P.

An alternative metric for comparison of distributions, the Bhattacharrya overlap, has recently been introduced to the vision community [63]. Effectively it correlates the pdfs of the two distributions after correcting for the variance, so that in our discrete case it has the value

$$\sum_{j=1}^{n} \sqrt{H_j P_j}$$

In order to test its significance, we convert it to its chi-squared form, the Matusta distance:

$$\chi_B^2 = 4 \sum_{j=1}^{n} (\sqrt{H_j} - \sqrt{P_j})^2$$

This metric was tested as an alternative to the chi-square distribution test but was found to yield poorer results (see Figure 5.6).

5.3.4 Determining the Actual Distribution

To enable use of the distribution comparison metrics, we must know the expected distribution of our measure. To this end we performed a Monte-Carlo simulation of the fitting process and recorded the results. We modelled the sensor noise process as a Gaussian plus quantization, which is an appropriate model for the laser range finder in use in our laboratory.

The distributions (graphed in Figure 5.2) were calculated as follows: For a given number of points n, the line $y = x + 1, x = 1...n$ was corrupted by Gaussian noise of $\sigma = 5$, then quantized to the next lowest integer. The runs histogram was calculated using the residuals of a linear least-squares fit. Repeating this process 5000 times, and measuring the cumulative frequencies for each length of run gave the distributions shown. This technique was chosen because it was felt that the particular choice of this line would not alter the results. To test this conjecture, the line slope and noise were varied widely and the experiment repeated. Results were comparable to within about 0.2 percent. However, changing the model to a quadratic altered the frequencies by up to 10 percent, suggesting that in real applications, it is important to 'train' the test on the models expected.

We note that although the histogram should be calculated for all possible values of the number of data points n (up to 10^6 in a 2D system), there was no significant change in the frequencies after about $n = 100$, lightening the computational load significantly.

5.4 Experiments

Two experiments were performed to assess the performance of the new test and compare it to existing tests. The experiments were designed to be representative of 'everyday' vision tasks: tracking and segmentation.

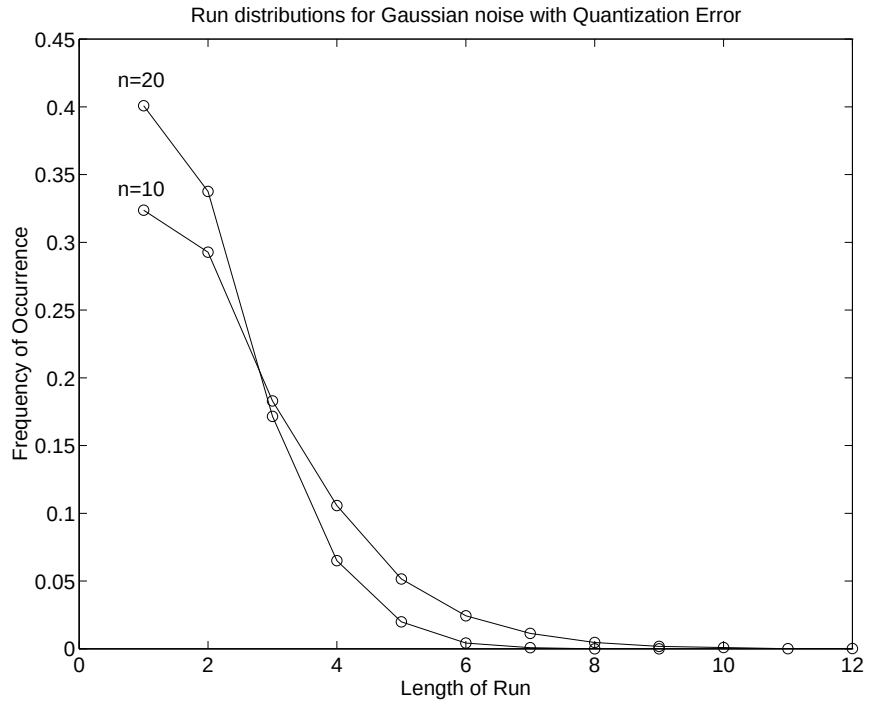


Figure 5.2: **Empirically derived distributions of run frequencies** for two values of n, the number of data points. The histograms were determined by counting the run frequencies from Monte-Carlo simulation of a line fitting procedure. These are then used as the base distribution with which test cases are compared.

5.4.1 Tracking

Here we consider the problem of tracking a point through time or space while maintaining an estimate of its trajectory. The tracking can often be foiled when one point passes in front of another and the program begins to follow the second point. The error may be detected by examining the fit between the trajectory model and the data. In this experiment the track is represented by a line at 45 degrees which proceeds for 50 points. The false trajectory is then represented by a second line of 50 points joining the first at an angle of 90 degrees. Although the choice of $90°$ may seem arbitrary, using smaller angles proved to be equivalent to increasing the noise level on the $90°$ case. An example track is shown in Figure 5.3.

Procedure

The following experiment was performed 1000 times for each noise level:

1. The trajectory data $\{\mathbf{x}_i\}_{i=1}^{100}$ were generated as

$$\begin{aligned} x_i &= i \\ y_i &= \begin{cases} i & i \leq 50 \\ 100 - i & i > 50 \end{cases} \end{aligned}$$

2. Gaussian noise of standard deviation $\sigma \in 1,4$ was added to the y coordinates of trajectory described above. Noise was added only in the y direction in order that the

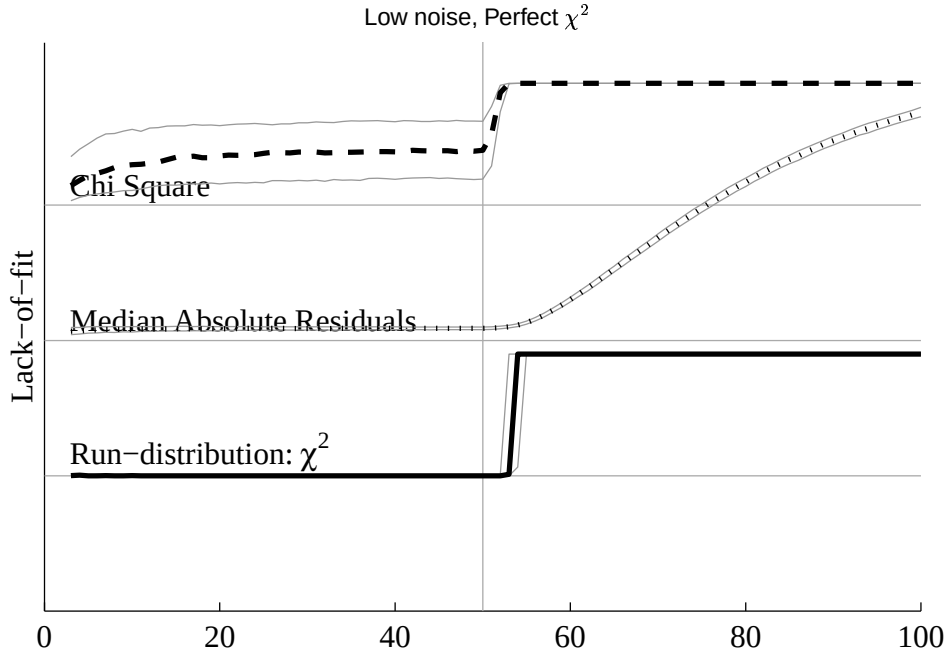


Figure 5.3: **Tracking task: Average-case performance**. The bottom of the figure shows the noisy data set. The three upper graphs are mean response curves of the lack-of-fit detectors as a function of position along the curve, with gray curves indicating the 25% and 75% percentile responses over the 1000 runs. The ideal detector will have low response on the left of the vertical line, with high response immediately afterwards.

χ^2 test can be applied in its most usual form. This is consistent with sampling of an unknown variable y at discrete time intervals x.

3. For each n between 3 and 100, the line $y = ax + b$ was least-squares fitted to the noisy data points $\{\mathbf{x}_i\}_{i=1}^{n}$ and the computed statistics of each of the goodness-of-fit tests were recorded. Again, we do not use orthogonal regression, as the noise model has errors only in y.

This generates 1000 traces of 97 response values for each of the three tests. We expect the tests to accept the model when $i \leq 50$ ("before the breakpoint") and reject it when $i > 50$. The experiment is varied for different values of the noise level, *and* for different values of the χ^2 test's noise parameter.

Results

To combine these results, we must condense the many traces into a form that accurately summarizes the relative merits of the test.

Figure 5.3 shows the mean response of each statistic over the runs taken with low noise and where the χ^2 test has perfect knowledge of the sensor noise level. We immediately note that the chi-squared trace is significantly higher than the others in the region before the breakpoint, but that it nevertheless shows a sharp increase at the breakpoint. This sharp increase means that a threshold can be found which will correctly distinguish the two tracks. In fact, in general, the chi-squared is thresholded at quite a high value, often greater than 0.99.

With the RD test, the choice of such a threshold is more obvious—the test responds more sharply and can therefore be thresholded at 0.5. However, all thresholds which cause

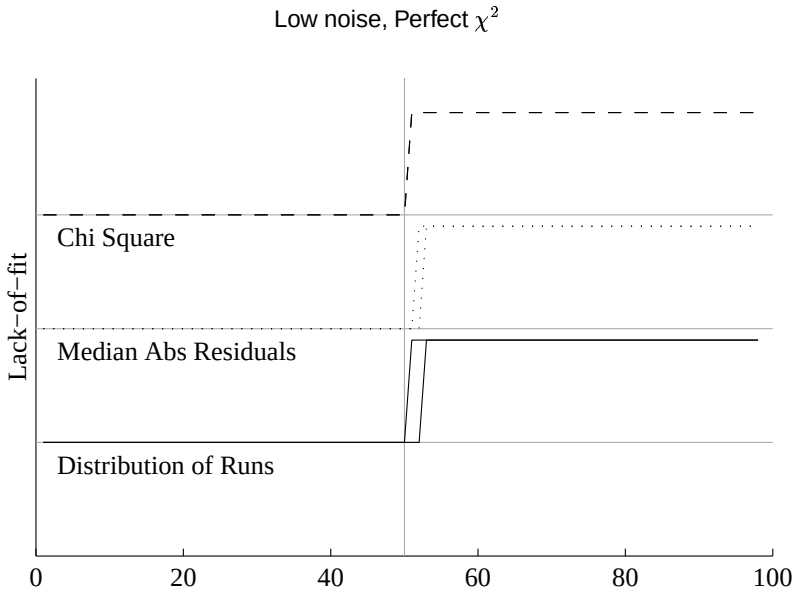
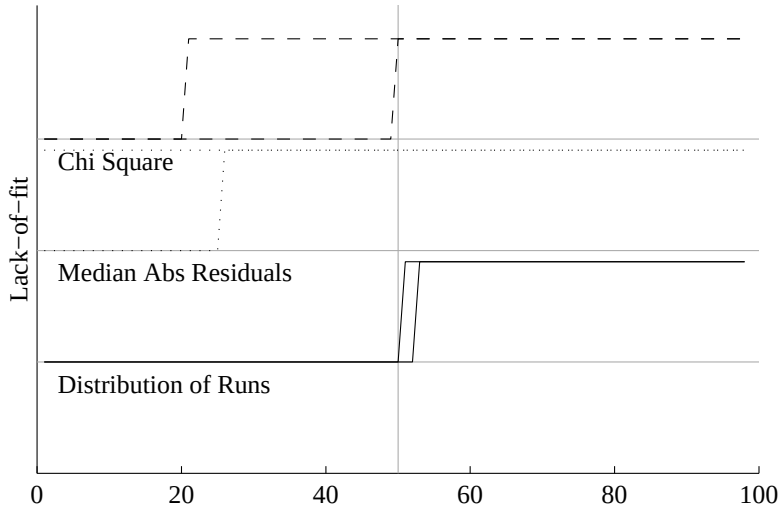


Figure 5.4: **Tracking task: Percentile performance**. For each detector two response curves are plotted, after thresholding at a 99% confidence level (or at the expected value for the MAD). The leftmost breakpoint corresponds to the point at which the test begins to reject the model, the rightmost to the point where it last accepts it. In this case the χ^2 test demonstrates near-perfect behaviour, while the RD test is delayed in responding.

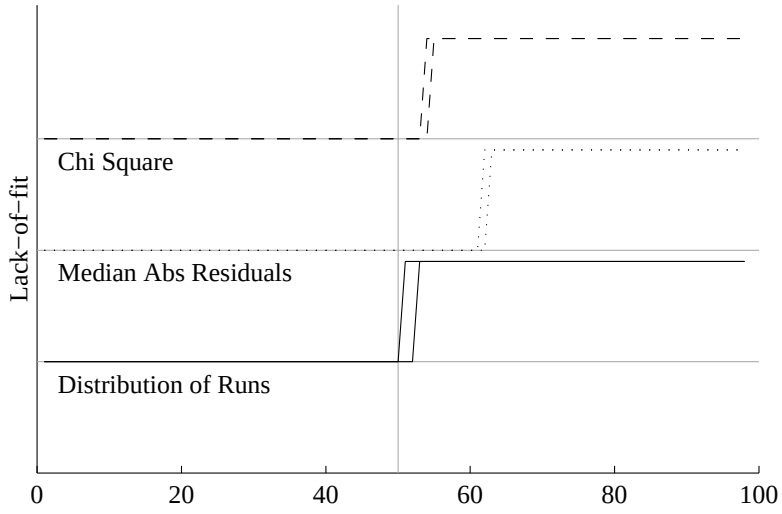
no false rejections will tend to delay the response at the breakpoint, introducing a delay in the classification. This delay means that the RD test is less sensitive than the chi-squared in this case. This is to be expected, as the χ^2 test is operating with the correct *a priori* noise estimate, while the run-distribution test is using no *a priori* information.

Although the mean response gives an indication of the relative merits of the algorithms, it is sometimes more useful to consider percentile responses. For example, it may be more important to know that an algorithm fails less than 10% of the time. To this end, Figure 5.4 shows the quartile responses of the tests, when thresholded at the 99% confidence level. These traces generate two sharp breakpoints—the first when more than 25% of the 1000 experiments are accepted, and the second when more than 75% are rejected. The width of the region between these two breakpoints is then an indication of the variation in results produced by the test, while the position of this region relative to the correct breakpoint indicates the sensitivity of the test. Again the RD test shows similar power to the chi-squared, but with a two to three sample delay.

Chi-squared with incorrect *a priori* noise level: Both the previous results demonstrate that the RD test comes close to the power of the χ^2 test, even though it has no *a priori* information. An interesting question is then: how accurately must we specify the noise parameter of the χ^2 test, in order that it outperforms the non-parametric runs test? Figure 5.5 shows results from the tests, with under and overspecified χ^2. We see that with the noise parameter underspecified by only 20%, the parametric (χ^2 and MAD) measures reject almost half of the correct tracks while, when the noise is underspecified, false acceptances occur until the breakpoint has moved considerably further along the track.



(a) Chi-squared given too low a noise parameter (80% of the true value). False rejections occur before the breakpoint is reached.



(b) Chi-squared given too high a noise parameter (150% of true). Detection of the false track is delayed.

Figure 5.5: **Tracking task: Inaccurate noise model**. The chi-squared and MAD responses are either too early (a) or delayed (b), compared with their superior performance given the correct noise level. The RD test response is of course the same for all experiments.

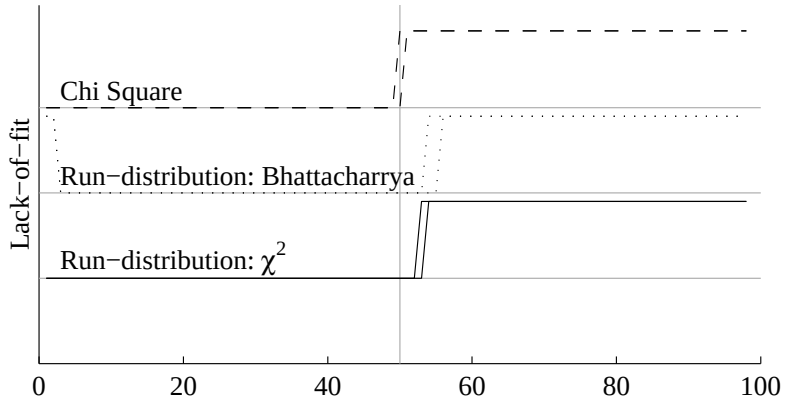


Figure 5.6: **Tracking task: Comparison with Bhattacharrya**. This track shows the results of comparing the distributions using the Bhattacharrya metric rather than the χ^2 in the runs test. For this task the Bhattacharrya metric introduces both a delay in response and false rejection for small numbers of points.

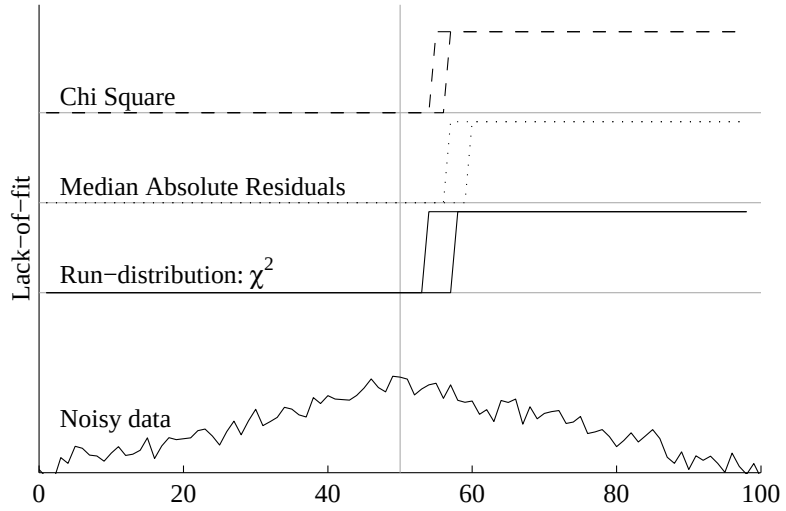


Figure 5.7: **Tracking task: High noise**. Quartile responses for the three thresholded tests, high noise. Even with the correct noise parameter, the χ^2 response is delayed with respect to the runs test. The RD test, however, has a greater variance in results.

Discussion

The experiment indicates that the RD test is approximately comparable to the parametric methods when we supply the correct noise level. This is in itself an interesting result, as the RD test has no tuning parameters. However, the real advantage of the RD test becomes apparent as noise is increased. The χ^2 test, with a slightly incorrect *a priori* noise model, quickly begins to perform more poorly than the new test. Finally, within the runs test, the question of whether to use the Bhattacharrya or χ^2 measures to evaluate the similarity of the run distributions is addressed in Figure 5.6. The χ^2 is superior in both accuracy and sensitivity.

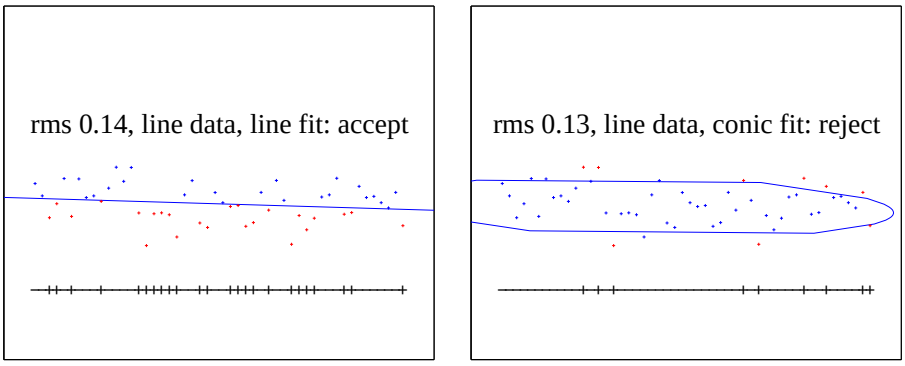


Figure 5.8: **Model selection.** Examining the run distribution also allows us to detect when a model of too high an order has been fitted, as the high-order curve will tend to induce structure into the residuals as it changes shape in order to fit the model. This example shows the result of the Kanatani conic fitter applied to the line data on the left. Although the fitting error has been reduced, a significantly different run distribution is produced. [Note that the peculiar shape of the conic is due to an error in the drawing program—the residual signs are correct with respect to the true equation.]

5.4.2 Segmentation

In order to evaluate the RD test on real data, it was applied to the problem of curve segmentation. Edgel chains from a local implementation of the Canny edge detector, and data from Rosin's archive [109] were used as input to a very simple greedy segmentation algorithm. The algorithm follows the boundaries produced by the edge tracking process and refits at every point. The residuals from the fit are submitted to the runs test, and a new segment is started if the test indicates that the model is a bad fit. In the implementation, there are a number of hysteresis strategies adopted in order to address the problem of breakpoints introduced by individually peculiar pixels. These are best documented by the pseudocode in Figure 5.15, but are summarized now. Essentially a segment is broken when it fails the RD test at the 99% confidence level, but such a break is accepted only when the segment fails at the 99.9% confidence level.

Model selection

The segmentation task does introduce the problem of model selection—given sampled data, and two possible models (in these tests, line and ellipse), how do we decide which model is appropriate. The *underfitting* case, where a line is used to describe an ellipse is the one which has been considered throughout this chapter, but it is the *overfitting* case which is often more interesting. In this case the residuals will certainly have a lower least square error, as the model has more freedom to represent the data. This generally leads to the introduction of MDL or AIC penalty terms (as discussed in Chapter 6). However, these terms depend again on the data noise level for their significance, so that their inclusion would somewhat prejudice this experiment's ability to characterise the performance of the run-distribution test. Fortunately the RD test itself can perform the model selection. Figure 5.8 illustrates how overfitting actually *introduces* structure into the residuals in its efforts to minimize the sum of squared errors. This then allows us to use the RD test to identify both over and underfitting in the data.

Results

Figures 5.9, 5.11, 5.12 illustrate the results on some sample datasets. The datasets exhibit quite different noise characteristics, and combinations of line and conic geometry. The set in Figure 5.9 is the output of subpixel interpolation after Canny edge detection. The edgels show an error level after fitting of about 0.1 pixels. While the general form of the results is similar to other reported schemes, there remain some deficiencies. In particular, the test is less sensitive than a correctly tuned χ^2 or MDL criterion, resulting in an overshoot on the breakpoints. Figure 5.10 shows a magnified view of the results, compared with results obtained by replacing the RD test with an MDL criterion. The MDL criterion used was to add λM to the χ^2 value, where M id the number of degrees of freedom in the model—2 for a line and 5 for an ellipse. Many values of λ were tried, and values in the range 0.05 to 0.25 pixels were found to give qualitatively the best results. The noise parameter required was varied from 0.05 to 0.4 pixels, and the results are shown for for typical values. As in the previous experiment, the RD test falls short of the best-tuned MDL performance (for $\sigma = 0.1$), but compares well with the mistuned results.

The dataset in Figure 5.11 contains pixel-accuracy edges, which means that quantization is the predominant influence on the error model. Despite the fact that this severely tests our requirement for a symmetric noise distribution (see Figure 5.13), the results are again comparable to those in the original paper. Finally, Figure 5.12 shows a synthetic dataset with a particularly odd noise distribution. In this case the RD test begins to break down, correctly segmenting the large-scale features but undersegmenting the fine-scale ones.

The tendency towards oversegmentation on this task may be due to the bias of the Ellipse-specific fitter. This tends to introduce additional structure into the runs on short segments, causing the test to reject short elliptical chains. This is despite the fact that the test will accept the larger chains of which the shorter ones are part. This may also be seen as a deficiency of the segmentation algorithm, which in its current form is particularly simple. Although the intention of this work is more to explore the utility of the RD as a parameter-free statistical test than to specifically address the curve segmentation problem, it would be interesting to discover if a more sophisticated segmentation algorithm would address some of the current deficiencies. In this case, another conic fitter might help despite the possibilty of returning hyperbolae.

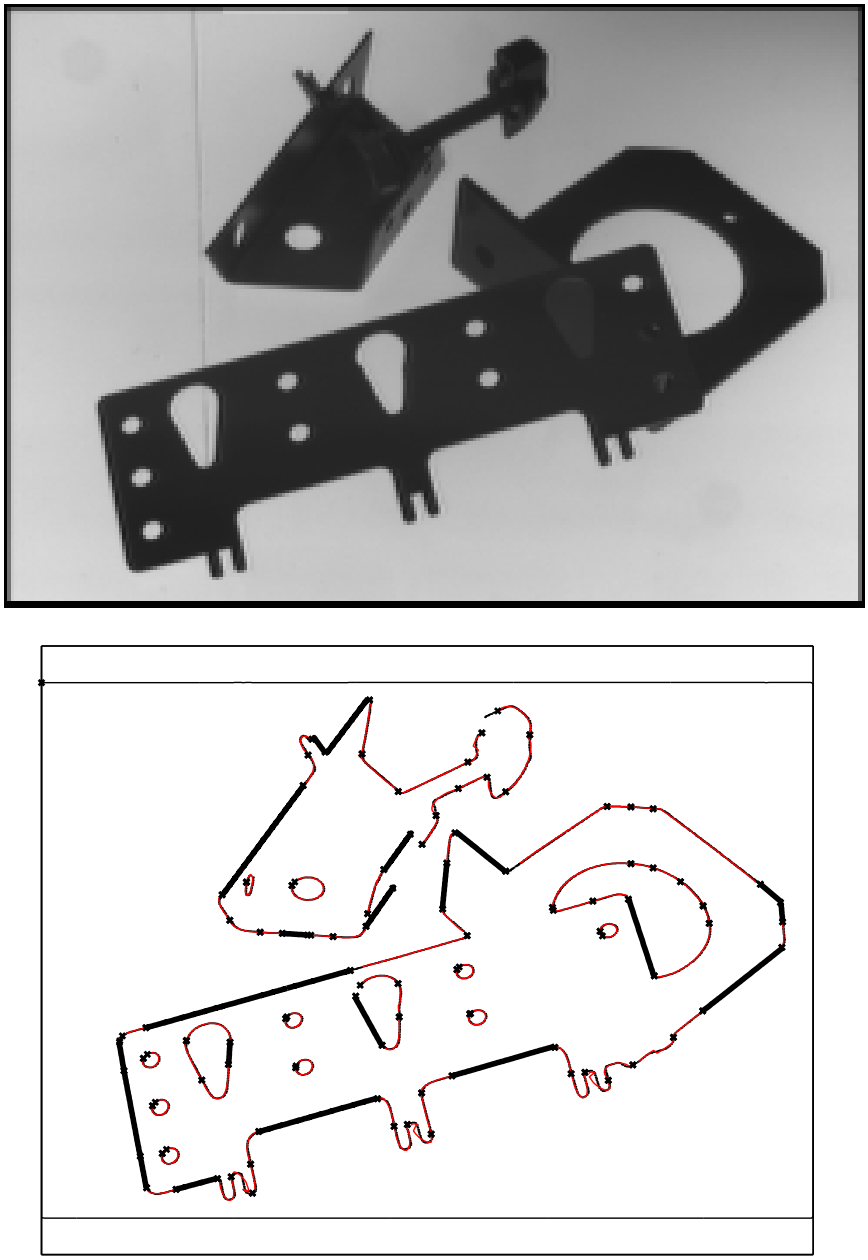
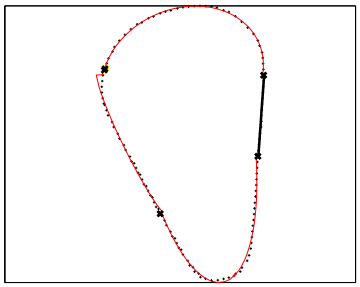
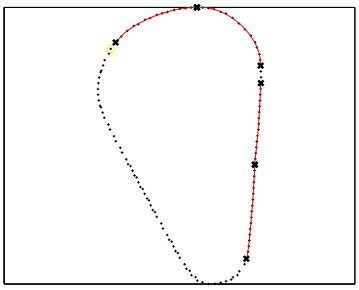


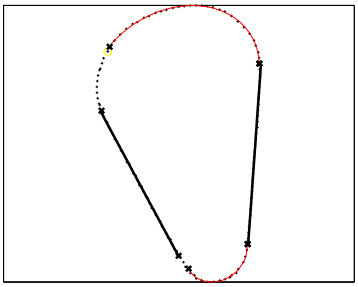
Figure 5.9: **Segmentation results: Subpixel edges**. Canny edgels from the image on the top have been segmented using the purely parameter free runs test. Lines are shown thicker than ellipses. Performance is qualitatively similar to standard techniques, but tends towards oversegmentation. Image kindly provided by C. Rothwell.



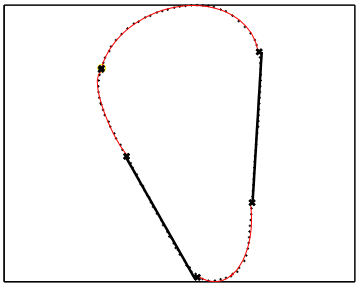
Run-distribution



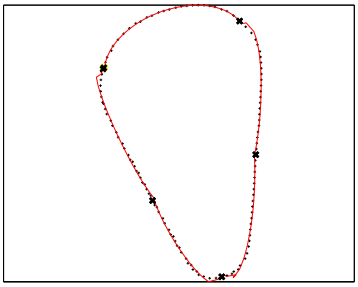
MDL, $\sigma = 0.05$



MDL, $\sigma = 0.1$



MDL, $\sigma = 0.2$



MDL, $\sigma = 0.3$

Figure 5.10: **Segmentation results: Subpixel edges**. Closeup of Rothwell image. Dots show the raw Canny edgels, thick lines the segmented lines and thin lines the segmented ellipses. The lower four segmentations are the result of the MDL criterion with the given estimates of noise standard deviation. The upper result is the output from the RD test. It is quite similar to the MDL with $\sigma = 0.2$ but the model selection has failed on the right hand edge.

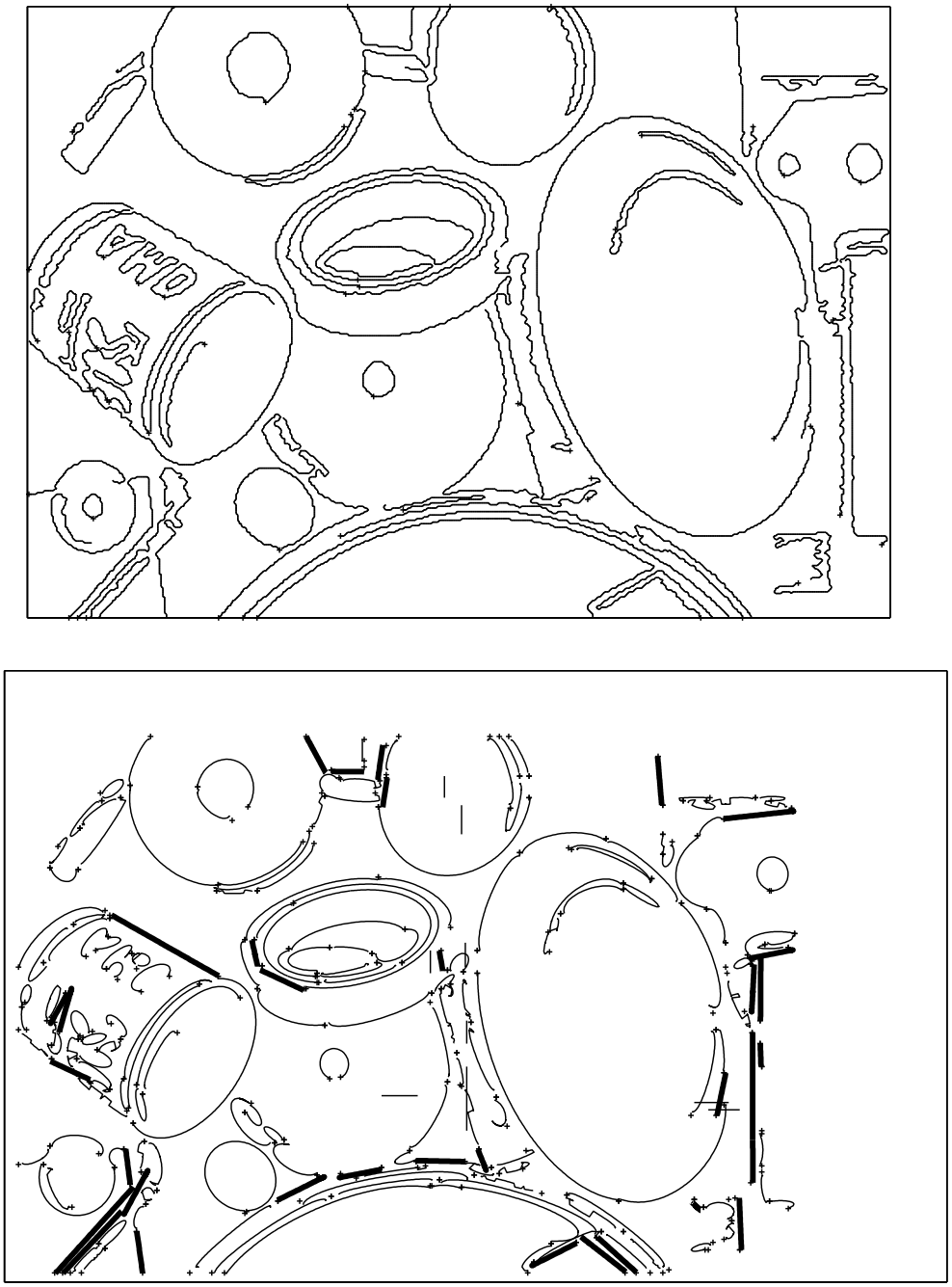


Figure 5.11: **Segmentation results: Quantized edges**. Only edges greater than 20 edgels in length have been fitted. Performance is qualitatively similar to [109], but there is again a tendency toward oversegmentation. Data supplied for FTP by P. Rosin.

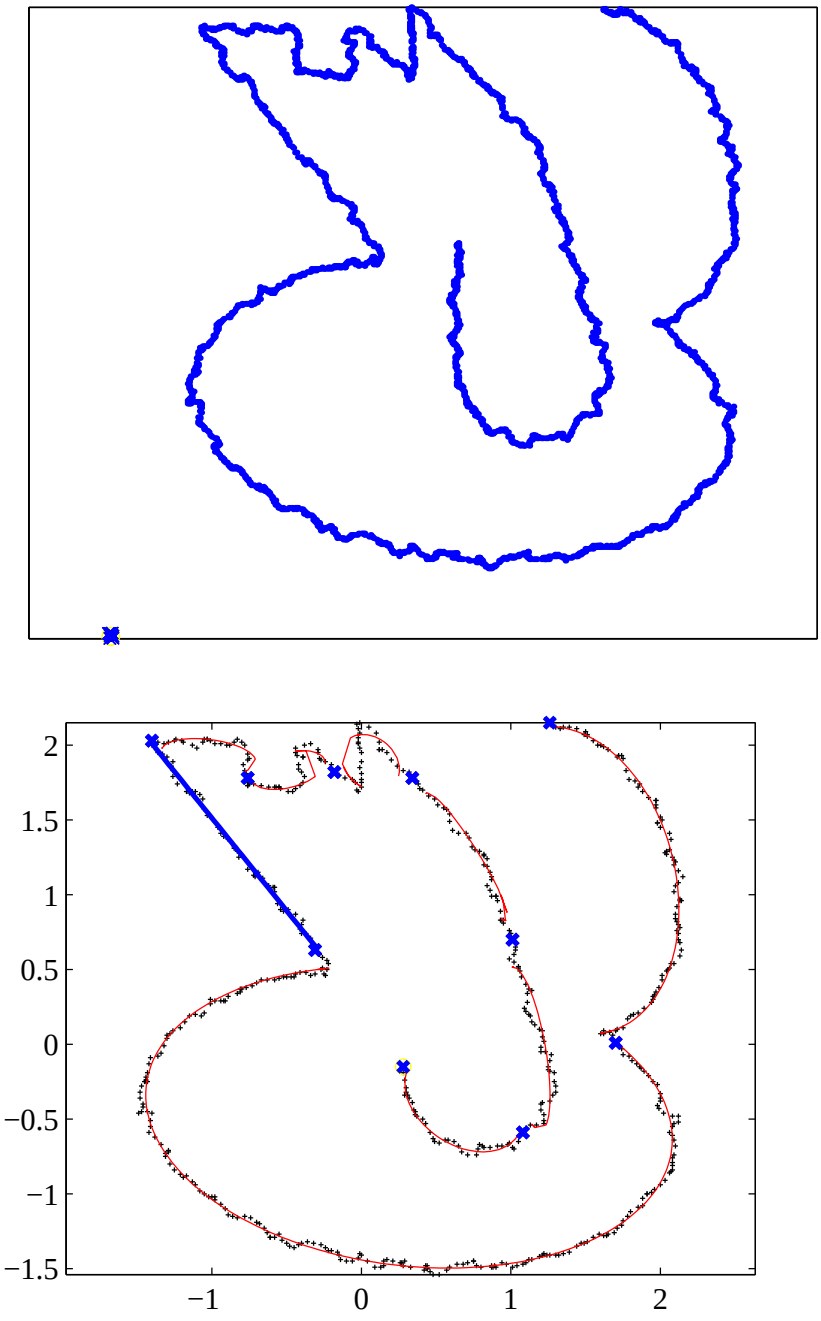


Figure 5.12: **Segmentation results: Synthetic noise model**. The final segment has not been reproduced due to an error in the segmentation algorithm. Data supplied for FTP by P. Rosin [109].

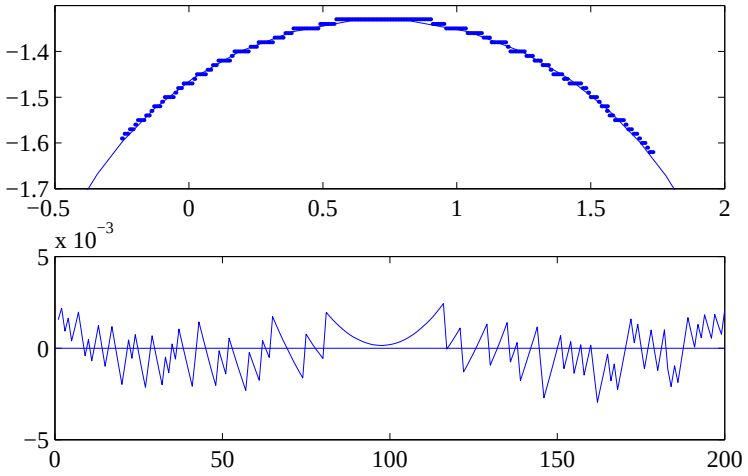


Figure 5.13: **Quantization error:** Demonstration of the residual structure induced by quantized data. The long straight section in the center of the ellipse produces an "ellipse-shaped" residual set. On short segments, this will lead to erroneous classification, although this particular segment is sufficiently long that the deviation is tolerated.

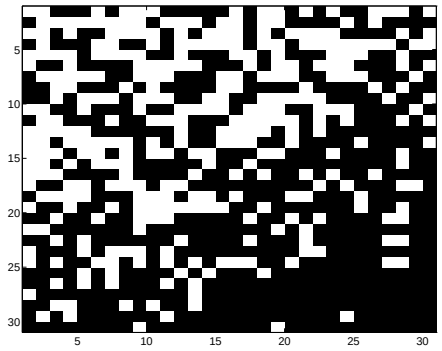


Figure 5.14: Example residuals sign map for a plane fit corrupted by several 10σ outliers clustered in the lower right corner. The 2D version of the test can define runs as connected components in the image.

5.5 Conclusions

We have introduced a new method of testing the hypothesis that some unknown data set is a noisy instance of a parametric model. Our method is superior to existing methods that make unrealistic assumptions about the noise characteristics of the input data. The method is fast, and can in most cases be made to have $O(n)$ time and space complexity. Sensitivity to small deviations in the model is high, while the false rejection rate is low, even when the data are heavily corrupted by noise. The major advantage of our test however is that there is no need to know the input noise level or distribution shape. On the conic fitting problem, the method is significantly cheaper than the chi-squared test, as the χ^2 must compute the geometric distance to each fitted point after fitting in order to calculate the statistic. Because the RD test needs only the sign of the distance, the much cheaper algebraic distance computation can be employed.

A problem with the test is that, although a very wide range of distributions is allowed, it is more sensitive than others to deviation from this class. For example, in situations where quantization error grossly exceeds sensor error, the noise model is violated and the false rejection rate increases sharply. Figure 5.13 illustrates this problem. This can be avoided by adding a little Gaussian noise to the data, but this is obviously not an ideal solution. The chi-squared test, in contrast, tolerates distributions which are non-Gaussian, as long as the noise parameter is tuned accordingly.

5.6 Areas for Future Work

The primary area for further work is in the development of higher dimensional versions of the test. [8] use a 2D analogue of the maximum run-length test, hinting at the definition of an n dimensional run: we assume that there is some topology defining adjacency between different data points – commonly the points are defined on a grid, implicitly providing such a topology. A run is then a connected set of points with the same label, the 'length' of the run becoming the volume of the connected set. Figure 5.14 shows an example 2D residual sign map, which illustrates the "runs" structure of a plane fit corrupted by outlier noise. Because of the higher probability of connectivity in higher dimensions, using area as the equivalent to 'length' of a run may need to be changed to a fractal measure of slightly lower dimension. This is currently implemented by using morphological operators to approximate the dimensionality reduction, and then measuring areas.

A secondary goal is to incorporate the RD test into a more sophisticated segmentation

- **Variables**
  - n Number of points
  - s, p Start and end indices of current segment
- **Parameters**
  - $l_{\min}$ Minimum segment length (7)
  - P_l, P_h Low and high confidence values (.99, .999)
- **Algorithm**

  while $s < n$

  Set breakpoint $p = s + l_{\min}$

  repeat

  Increment breakpoint $p = p + 1$

  Get segment $\mathcal{X} = \{x_i\}_{i=s}^{p}$

  Fit models (line,ellipse) to $\mathcal{X}$

  Select model with lowest RD score c.

  until $c > P_h$

  Move to last point p' at which c was $< P_l$.

  Report segment (s, p')

  Reset start point $s = p' + 1$

  endwhile

Figure 5.15: Pseudocode for the segmentation algorithm

algorithm, in order to more equally compare it to other metrics.

Chapter 6

Curve Segmentation using the Sum of Variance

This chapter introduces a new cost function for image segmentation. This metric is astonishingly simple: the total cost of expressing an image as a collection of nonoverlapping regions is the sum of the sample variances within each region. I give a theoretical demonstration that minimization of this metric correctly detects a step edge *without any a priori thresholds*, even in the presence of spatially varying noise. Results from simulation then illustrate that the technique is equally valid for piecewise constant segmentation of 1D signals. Although demonstrations are shown only for the curve case, the metric itself extends immediately to 2D—however the algorithms for its minimization do not extend so easily.

Additionally, when multiple segmentations are possible due to the presence of differing natural scales in the data, the natural segmentations are the only local minima of the metric as a function of number of breakpoints.

A simple implementation based on a genetic algorithm is used to apply the metric to real problems, and experimental confirmation of its performance for piecewise quadratic segmentation of laser stripe data is provided.

6.1 Introduction

Research into image segmentation has been a key theme in computer vision over the last thirty years. Even the simplest possible case—segmenting a one dimensional signal into piecewise constant regions—continues to attract interest. Of the many approaches considered in the literature, almost all depend to some extent on user-selected scale parameters in order to produce the segmentation.

In this chapter I introduce a technique of astonishing simplicity which can produce intuitively correct segmentations over arbitrary models, without the need for any user-specified parameters. Furthermore, if an image, such as the sine wave superimposed on the parabola in Figure 6.8, *does* have interpretations at a number of natural scales, these appear as local minima of the error metric as a function of the number of breakpoints.

If we express the segmentation task as an optimization problem, then the primary problem is reduced to identifying an appropriate fitness function. Expressing the problem more formally: given n image points $\{\mathbf{x}_i\}_{i=1}^{n}$, a *segmentation* $\mathcal{S} = \{R_m, \mathbf{a}_m\}_{m=1}^{M}$ is a covering collection of M model regions—disjoint subsets of the data points. A *region* R_m is a subset of the data points, and the *parameter vector* $\mathbf{a}_m$ is the vector of model parameters describing the region. In the $1D$ case, a convenient representation of the region uses the index i_m of the first datapoint within that region, so such a segmentation may be expressed as $\mathcal{S} = \{i_m, \mathbf{a}_m\}_{m=1}^{M}$. In addition, we will find it convenient to define $n_m = i_{m+1} - i_m$, the number of points in region m; the index function $m(i) := m$ such that $i_m \leq i < i_{m+1}$

n	Number of data points
M	Number of model regions
$\mathbf{a}_m$	Model parameters of region m
$i_m \in [1, n]$	Start index of region m
$n_m = i_{m+1} - i_m$	Length of region m
$m(i) \in [1, M]$	Index of region containing pixel i
$\mathbf{x}_i = (i, y_i)$	Data point i
$r_i = D(\mathbf{a}_{m(i)}, y_i)$	Residual of datum i

Table 6.1: Summary of notation used for the segmentation of 1D signals.

which maps points to regions; and the *residuals* r_i as the distance between point $\mathbf{x}_i$ and the model defined by $\mathbf{a}_{m(i)}$. This notation is summarized in Table 6.1.

Given an error function E defined over the space of all segmentations, the optimal segmentation is then that for which the global minimum of E is achieved. This chapter introduces the *sum of variance* metric

$$E(\mathcal{S}) = \sum_{m=1}^{M} \hat{\sigma}_m \tag{6.1}$$

where the sample variances $\hat{\sigma}_m$ are defined as

$$\hat{\sigma}_m = \frac{1}{n_m} \sum_{i=i_m}^{i_{m+1}-1} r_i^2$$

In the following sections I briefly review previous segmentation research, placing the sum-of-variance metric in the context of previous optimization based approaches, and motivate the choice of E as a segmentation metric. Theoretical examination of the simple case of piecewise constant approximation further clarifies the metric's behaviour. Simulated examples with nonuniform, *a priori* unknown noise are augmented with examples on real laser stripe data using an implementation based on a genetic algorithm. These real data examples illustrate that this system finds intuitively accurate segmentations. Even though the meaning of "good" is not well defined for the segmentation problem, we emphasize that these results are obtained with no thresholds or parameters. Current and future research opportunities are discussed.

6.2 Background

The two main approaches to segmentation may be classified under the headings of local feature detection and global model-based fitting. Local methods [98, 3, 77], based on analysis of the curvature function or some discrete approximation of it, generally require some image smoothing in order to remain robust to effects of noise. As smoothing may in itself be cast as a model-based segmentation process [20, 87, 114] we concentrate here on the problem of fitting piecewise continuous model segments to image data, while simultaneously identifying discontinuities.

6.2.1 Discontinuity-preserving minimization

Adaptive smoothing [114, 119], weak plate [11, 67], and MDL (minimum description length [85, 66]) techniques all amount to minimizing a cost function which is a tradeoff between "truth": the sum of residual errors; and

"beauty": the model smoothness and number of discontinuities. In the notation above, the model parameter vector $\mathbf{a}$ is a discrete representation of the image $\mathbf{a} = \{u_i\}_{i=1}^n$ and the error looks like

$$E = \sum_{i=1}^{n} \phi(r_i) + \psi(M) + \omega(\mathbf{a})$$

where ϕ measures the deviation of the model from the data points, ψ is a function which favours "simple" models (i.e. those with a small number of segments) and ω measures the model smoothness or simplicity. Most commonly, these influence functions are simply linear or quadratic and we can assume that once the breakpoints are fixed minimization is trivial. Then the core minimization is over an error of the form

$$E = \sum_{i=1}^{n} r_i^2 + \lambda_1 M + \lambda_2 \|\nabla^2 \mathbf{a}\| \tag{6.2}$$

where the λ_k are tuning parameters which are set to indicate a prior preference for smoothness or simplicity over fidelity of representation. While such techniques yield a natural scale-space, they provide no information as to preferred choices of λ.

The implementation of [114], which will be explored in this chapter, approximates the anisotropic diffusion algorithm of [87] using an iterative technique. Their algorithm is iterative, and at each iteration each image point $\mathbf{x}_i$ is replaced by

$$\mathbf{x}_i' = \frac{\sum_{j=i-1}^{i+1} w_j \mathbf{x}_j}{\sum_{j=i-1}^{i+1} w_j}$$

where the weights w_j are inversely dependent on the local gradient magnitude at the point. For a 1D signal $\mathbf{x}_i = (i, y_i)$ this weight is

$$w_j = \exp\left\{ -\left(\frac{|y_{j+1} - y_{j-1}|}{\lambda} \right)^2 \right\}$$

In the experiments reported here, smoothing was stopped when the relative root-mean-square difference between successive iterates was less than 10^{-4}. Although this algorithm does not explicitly segment the data, it has the effect of smoothing the data so that a very wide range of simple step-edge thresholds will identically do so. In this sense it performs the vast bulk of the segmentation task.

6.2.2 The MDL criterion

The standard MDL criterion [101, 66, 1] derives its error metric from Occam's razor and is based on the consideration that the most natural description of a data set is that which is the most simple. In information-theoretic terms, this corresponds to the description which requires the fewest bits to encode. Following Leclerc's formulation, the cost of representing the image is the encoding cost of the models *plus* the cost of Huffmann-encoding the residuals. The cost of the residuals is then

$$-\log_2 \prod_{i=1}^{n} P_i$$

where P_i is the probability that the residual r_i is consistent with being sampled from the noise distribution. In the piecewise-constant, Gaussian noise case, this yields the overall cost

$$E_{\text{MDL}} = \sum_{i=1}^{n} r_i^2 + \lambda M$$

The cost per segment λ is derived from the *a priori* specification or estimation of the image noise distribution, and therefore in the case where the noise is not accurately known, it becomes a tuning parameter. Effectively, then, the cost of encoding a segment of length n_m is equal to the sum of the n_m residuals plus the number of model parameters times the cost per parameter. MDL has an advantage over the smoothing-based approaches in that the value of λ is prescribed, but only when given accurate noise pdfs and image complexity priors.

6.2.3 Parameter-free techniques

Of the many image description techniques proposed in the literature, very few are completely free of tuning parameters. One notable exception is Lowe's line segmentation algorithm [72, ch 4], and its recent extension to curves [109]. The Lowe algorithm defines the *significance* of a region of data points as the ratio of the maximum deviation of the points from the line segment interpolating the endpoints to the region size, or

$$E_{\text{Lowe}}(m) = \max_{i_m \le i < i_{m+1}} \frac{|r_i|}{n_m}$$

The algorithm then decides between the zero and single-breakpoint interpretation of a segment by comparing the significances of both interpretations and choosing the lower of the two. In this sense, Lowe's algorithm may be viewed as an L_∞ analogue of the minimization of sum-of-variance proposed in this chapter. However, as presented, the class of polylines which the algorithm fits is limited by the requirement that the polyline interpolate a subset of the data points. In addition, the use of maximum deviation renders the method susceptible to outliers.

6.3 From MDL to Sum of Variance

This section explores the relationship between MDL and the new metric, showing how some philosophical or conceptual difficulties with MDL can lead to sum-of-variance for problems where the model complexity does not vary across the image.

6.3.1 What is the cost of the noise?

Examining the MDL cost reveals that it is dominated by the cost of Huffman encoding the *particular set of noise residuals in the image*, when in fact all we want to do is encode the underlying representation and record the parameters of the *distribution* from which those residuals were sampled. Put another way; in order to reconstruct the original image we will be content simply to reconstruct the model and to add noise sampled from the same distribution as the original image.

If the underlying distribution is zero-mean Gaussian, then the only parameter of the distribution is its variance. To cope with spatially-varying noise, the variance must be evaluated in each region of the segmentation, yielding the metric

$$E(\mathcal{S}) = \sum_{m=1}^{M} \hat{\sigma}_m + \lambda M \tag{6.3}$$

where the sample variances $\hat{\sigma}_m$ are defined as before:

$$\hat{\sigma}_m = \frac{1}{n_m} \sum_{i=i_m}^{i_{m+1}-1} r_i^2$$

6.3.2 The choice of prior model distribution

The second difficulty with MDL comes in the choice of description language. Suppose we wish to represent as efficiently as possible a particular segmentation $\mathcal{S} = \{i_m\}_{m=1}^{M}$ of the data. In the case where we expect few segments, a seemingly obvious representation might be the list of breakpoints $\{i_m\}_{m=1}^{M}$, with encoding cost $\sum \log_2 i_m$ bits. However we cannot simply use this cost, as it favours segmentations with smaller i_m. The cost for each i_m must be the cost of representing any number between 0 and n, i.e. $\log_2 n$, yielding the cost $M \log_2 n$.

However, suppose that we expect a reasonably large number of segments—say we have $n = 128$ and expect about 20 segments. Then the total cost of encoding 20 segments is $20 \log_2 128 = 140$ bits. In this case it would be cheaper to represent the segmentation by the boolean vector $\{b_i\}_{i=1}^{128}$ in which $b_i = 1$ only where $i = i_m$ for some m. This then tends to argue that the cost for all models ought to be 128 bits, which is consistent with a completely general prior. If we take this approach, the MDL metric then reduces to

$$E(\mathcal{S}) = \sum_{m=1}^{M} \hat{\sigma}_m + 128 \tag{6.4}$$

Discarding the constant term which applies to all models, we are left with the sum-of-variance metric.

6.4 Experiments

Before theoretically motivating the behaviour of the metric (6.1), we present some experiments that compare its performance with that of MDL and an adaptive smoothing algorithm on data with nonuniform noise.

6.4.1 Implementation of search

As we are comparing the metrics rather than any algorithm's ability to minimize them, we wish to ensure that we get as close as possible to the global optimum. In the current implementation, a genetic algorithm (GA) similar to that used by Roth [110] was employed. Individual segmentations are represented by a boolean n-vector (where n is now the number of data points), with a 1-bit indicating a new segment starting at that point. The standard crossover operator is used, with reproducing pairs chosen with a bias toward higher fitness. Mutation is a combination of low-likelihood bit reversal, and a Gaussian variation of the breakpoint positions.

In order to verify the optimizer's ability to find global minima of the cost functions, we performed exhaustive search of all segmentations with fewer than 6 breakpoints for a number of synthetic 2-to-4 breakpoint problems. In all cases the GA identified the optimum in under a minute. While this experiment gives no guarantee that the GA will do so in other cases, it does inspire some confidence in the algorithm's results.

In implementation of the sum-of-variance algorithm we replace the sample variance $\frac{1}{n}\sum_{i=1}^{n} r_i^2$ of the residuals, with the unbiased variance estimate $\frac{1}{n-p}\sum_{i=1} R_i^2$, where p is the number of degrees of freedom of the model. In the piecewise constant case, this is 1, for the piecewise quadratic model $p = 3$. As well as being statistically more justifiable [59], this has the generally useful effect of eliminating from consideration the interpolating segmentations—at least one extra point must be available to estimate the variance. In practice, we fit lower-order models to the short segments, thus disallowing only the single point per segment interpretation.

This however, is an opportunity to influence the segmentation produced. It might be argued that one requires an extra data point (over the minimum required in order to fit the

model) to compute the variance. In this case p, the number of degrees of freedom (d.o.f.) could be increased to 2 for the piecewise constant case, and 4 for the quadratic case. In effect, this introduces a boolean tuning parameter into the algorithm, which will be shown to have a small effect on the segmentations produced.

6.4.2 Comparison of MDL, SOV and adaptive smoothing

Figure 6.1 shows the raw data used for this experiment: a piecewise-constant signal is corrupted with spatially-varying Gaussian noise, simulating an image of two differently textured surfaces. This signal is difficult as the step discontinuity in the low-noise region has a significantly lower height than the noise standard deviation in the high-noise areas.

Figure 6.1 also shows the minimum cost sum-of-variance segmentation which is correct with respect to the ground truth. The results for the standard MDL metric, and the Saint-Marc and Medioni [114] adaptive smoothing implementation of Perona-Malik anisotropic diffusion [87] are shown in Figures 6.2 and 6.3 respectively, for different values of their tuning parameters λ. Note that, as these algorithms assume a constant noise level over the image, there is no value of their tuning parameters that can extract the original model. This is demonstrated by the fact that the low-noise step ceases to be segmented *before* the high-noise region is successfully marked as flat. Figure 6.4 shows further examples with uniform noise and varying noise with a constant signal.

6.5 Theoretical motivation

In contrast to the standard MDL metric, it is not immediately easy to see why the metric (6.1) should give the correct results that the previous experiments indicate. Fortunately, the simplicity of the metric allows us to provide some theoretical motivation for the behaviour of its minima. This section explores the simple case of a step-edge signal, and computes the value of (6.1) for the main possible segmentations. In observing how the metric changes as we under- or over-segment the signal, the behaviour of the metric becomes clear. Let us consider the simplest case of a step edge of height h, corrupted by Gaussian noise of variance σ, represented by

$$y_i = r_i + \begin{cases} h & 0 < i \leq n \\ 0 & n < i \leq 2n \end{cases}$$

where the residuals r_i are drawn from a zero-mean Gaussian process (Figure 6.6 top). For this exposition we assume that n is sufficiently large that we may take expectations over sums of the residuals, and we additionally require that individual segments contain sufficient points to accurately estimate σ. The relaxation of these assumptions is addressed later.

In this case, segmentations are represented cleanly by the *line process* [11] $\mathcal{S} = \{i_m\}_{m=1}^{M}$. The model parameters $\mathbf{a}_m$ are simply the means of the region points, allowing us to quickly calculate E_{SOV} for various segmentations.

Cost of the null segmentation

The cost of the null segmentation $E_{\text{SOV}}(\{\})$ is found by first calculating the sample mean of the entire dataset

$$\begin{aligned} \hat{y} = E[y] &= \frac{1}{2n}\Big(\sum_{i=1}^{n}(r_i + h) + \sum_{i=n+1}^{2n} r_i\Big) \\ &= \frac{1}{2n}(nh) + E\Big[\sum_{i=1}^{2n} r_i\Big] \\ \hat{y} &= h/2 \end{aligned}$$

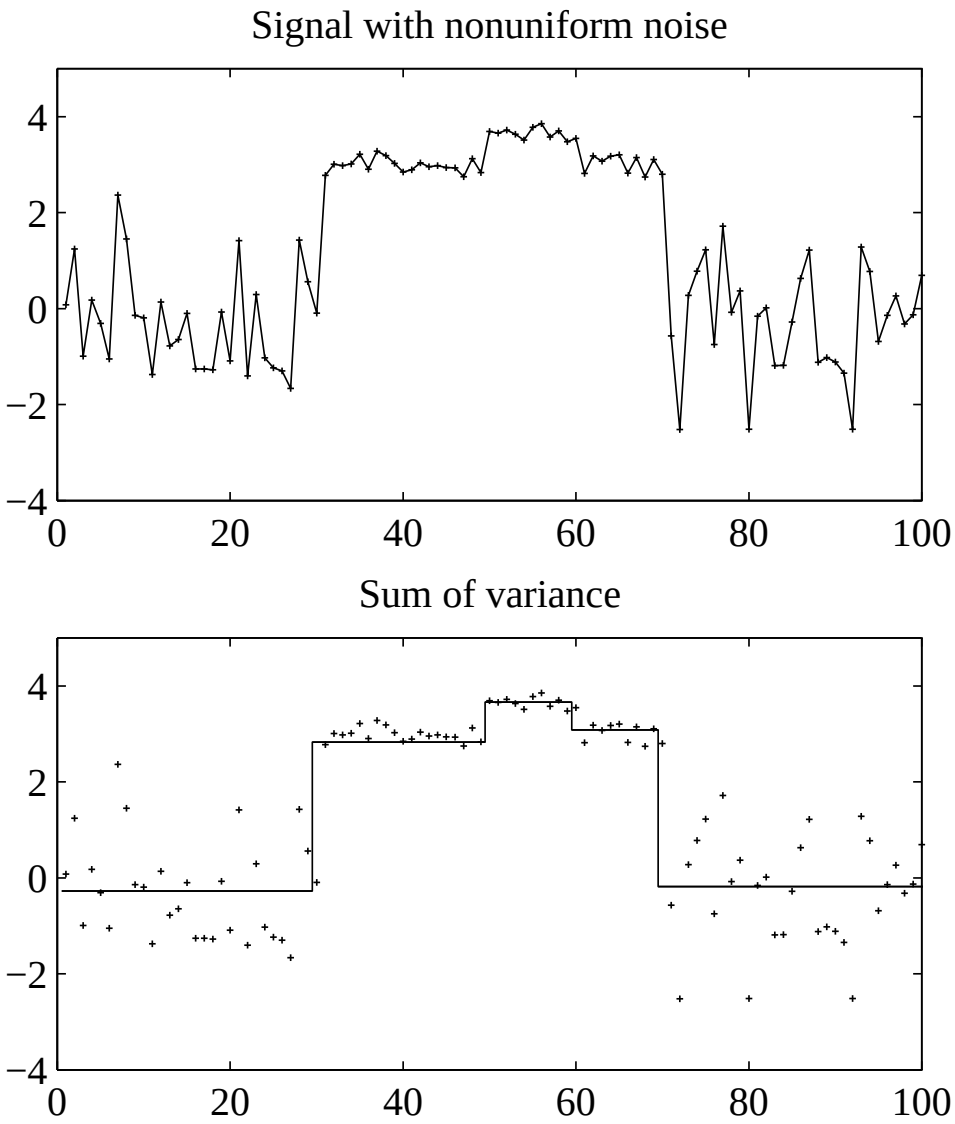


Figure 6.1: The data used in the comparative tests and its sum-of-variance segmentation. The segmentation produced by sum-of-variance places the breakpoints at exactly the positions of the ground-truth segmentation.

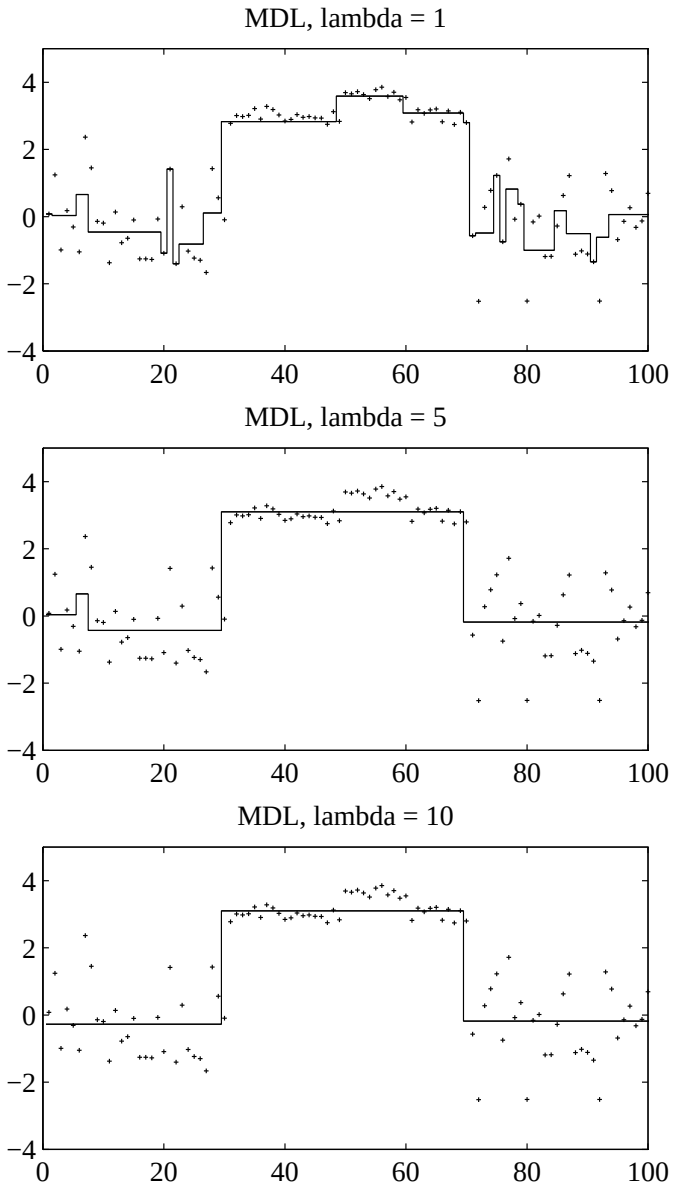


Figure 6.2: Standard MDL. The globally minimum segmentation of $E_{\text{MDL}} = \sum_{i=1}^{n} r_i^2 + \lambda M$ is plotted for three values of the noise parameter λ. No value gives the ground-truth interpretation.

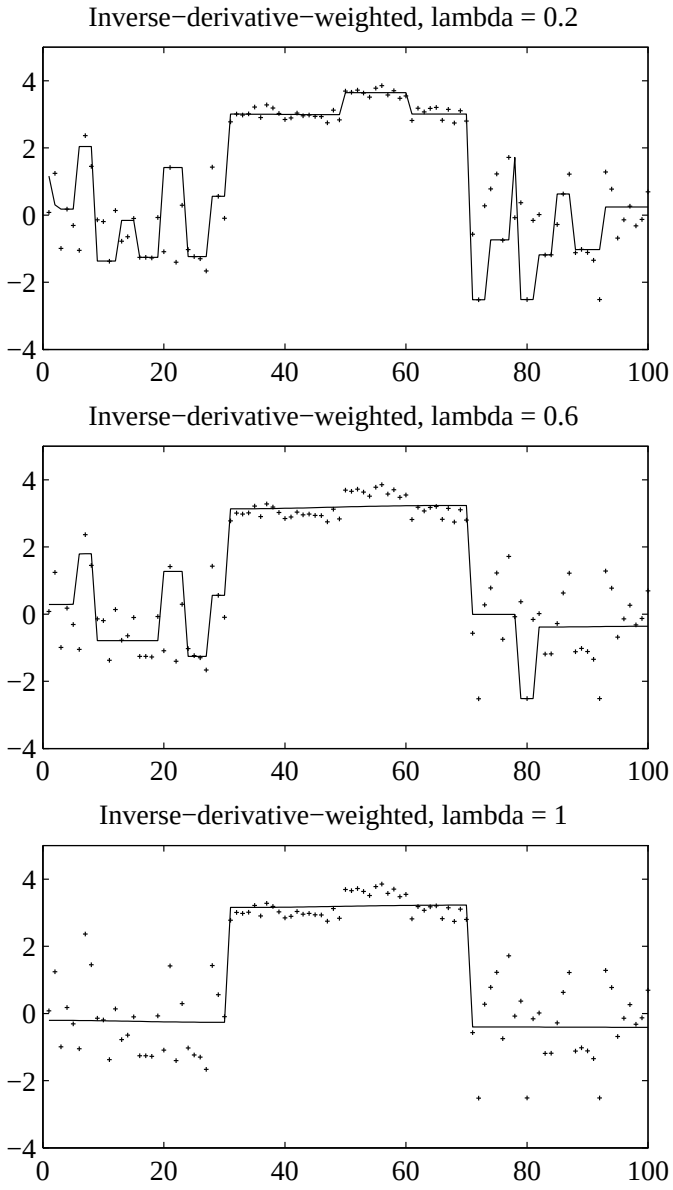
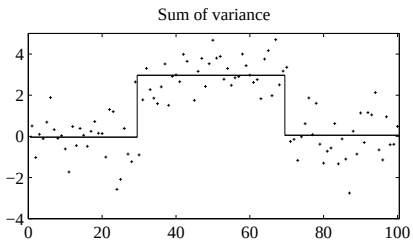
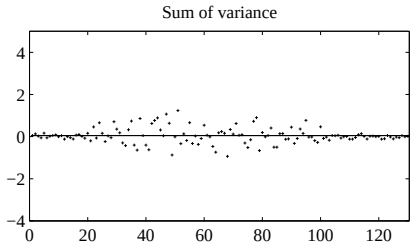


Figure 6.3: Results of Saint-Marc & Medioni adaptive smoothing. The results are plotted for three values of the weighting parameter λ. No value gives the ground-truth interpretation.



Roof signal, uniform noise. Correct segmentation.



Constant signal, varying noise. Null segmentation.

Figure 6.4: Further SOV examples.

Where the sum of all residuals $\sum_{i=1}^{2n} r_i$ has an expected value of 0 for symmetric noise. The sum of sample variance is then

$$\begin{aligned} E_{\text{SOV}}(\{\}) &= \frac{1}{2n}\left(\sum_{i=1}^{n}(r_i + h/2)^2 + \sum_{i=n+1}^{2n}(r_i - h/2)^2\right) \\ &= \sigma^2 + \left(\frac{h}{2}\right)^2 \end{aligned} \tag{6.5}$$

Where the mean-square residual $\frac{1}{2n}\sum_{i=1}^{2n} r_i^2$ is replaced by its expected value σ^2 for Gaussian noise.

Cost of the single-breakpoint segmentation

Similarly, we may calculate the error for the correct single-breakpoint segmentation $E_{\text{SOV}}(\{n\})$ for which the left and right means are given by

$$\begin{aligned} \hat{y}_l &= h \\ \hat{y}_r &= 0 \end{aligned}$$

and the sum of variances is

$$E_{\text{SOV}}(\{n\}) = \sigma^2 + \sigma^2 = 2\sigma^2 \tag{6.6}$$

Cost of multi-breakpoint segmentations

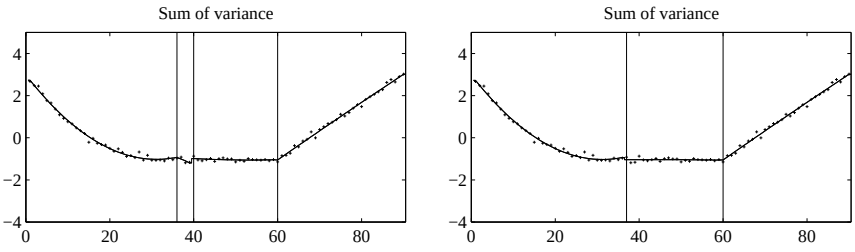
Finally we consider a two-point segmentation $\{r, n\}$, which is given by inserting a new breakpoint into the correct segmentation $\{n\}$. Then, the expected values of the variances are all σ^2, yielding

$$E_{\text{SOV}}(\{r, n\}) = 3\sigma^2$$

From this we see that adding additional breakpoints to the single-breakpoint solution must of necessity increase the value of E_{SOV}, as each segment will contribute σ^2 to the sum.

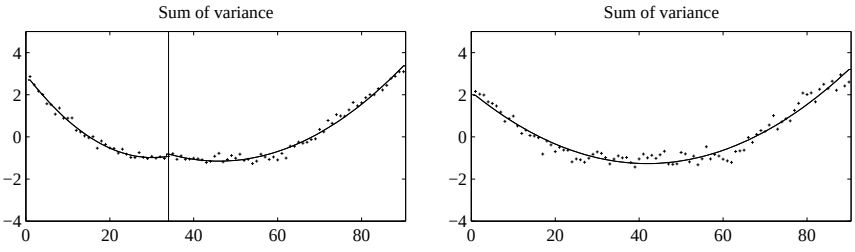
6.5.1 The minimum-cost segmentation

The segmentation which minimizes (6.1) is therefore either the step-edge or the constant solution. Which of these will be chosen depends on the relative values of h and σ. From



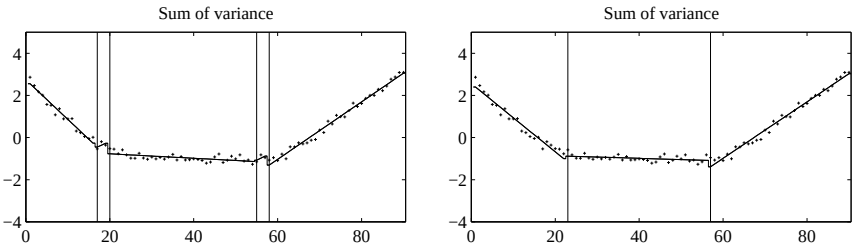
d.o.f. = 1. d.o.f. = 2.

Oversegmentation for d.o.f. = 1. Correct number of segments for d.o.f. = 2. In both cases the leftmost breakpoint (at the C_2 discontinuity) is poorly localized.



d.o.f. = 2. $\sigma = .3$ d.o.f. = 2. $\sigma = .5$

Higher noise levels. The ground truth segmentation was not reached using any d.o.f. between 1 and 5.



d.o.f. = 1. d.o.f. = 2.

Piecewise linear segmentation.

Figure 6.5: Piecewise smooth signal, uniform noise. SOV is run with a piecewise quadratic model. SOV oversegments when the degrees of freedom in the variance calculation is set to 1, and segments correctly when set to 2.

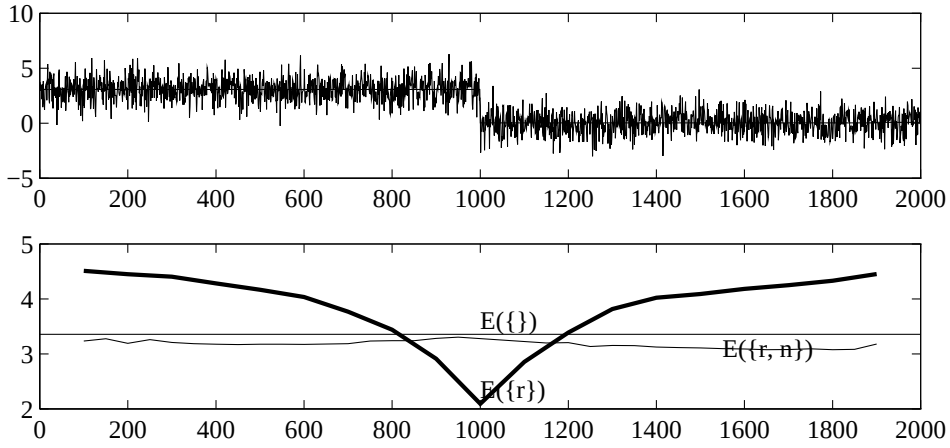


Figure 6.6: Experimental verification of the theoretically predicted values of SOV on piecewise constant data. The value of the null segmentation $E_{\text{SOV}}(\{\})$ is plotted as a horizontal line. The value of the single-breakpoint segmentation $E_{\text{SOV}}(\{r\})$ is plotted for values of r at every 100^{th} sample (thick line). Finally, the cost $E_{\text{SOV}}(\{r, n\})$ of augmenting the step-edge segmentation with another point is the slightly erratic thinner line.

(6.5) and (6.6), we will detect the step edge if

$$\begin{aligned} E_{\text{SOV}}(\{n\}) &< E_{\text{SOV}}(\{\}) \\ 2\sigma^2 &< \sigma^2 + \left(\frac{h}{2}\right)^2 \\ 2\sigma &< h \end{aligned}$$

This indicates that when the noise standard deviation exceeds half the height of the step edge, it will not be detected.

It is worth noting that a similar, although algebraically messier, analysis obtains when the breakpoint is not at the midpoint of the segment and when the noise is nonuniform over the segment.

6.5.2 Experimental confirmation

In order to experimentally verify the above argument, we generated a 2000-sample step edge of height 3, corrupted with Gaussian noise of unit variance. Figure 6.6 illustrates the sum-of-variance measure for three families of segmentations, showing that the step-edge interpretation at the correct breakpoint is indeed the cheapest model, by the predicted margin $(h/2)^2 + \sigma^2$.

6.5.3 Scale-space interpretation

One of the immediately striking properties of the sum-of-variance measure (6.1) is that it achieves its global minimum of zero for the interpolating segmentation $\{k\}_{k=1}^{n}$. How may we claim then that the minimum cost corresponds to the step edge segmentation $\{n\}$? The answer is that without *a priori* knowledge, both are equally valid interpretations—the data might really be a zero-noise sampling of a noisy surface.

This is reflected in the local behaviour of E_{SOV} as a function of the number of breakpoints. Figure 6.7 plots the value of E_{SOV} for the step edge as the spacing of the breakpoints

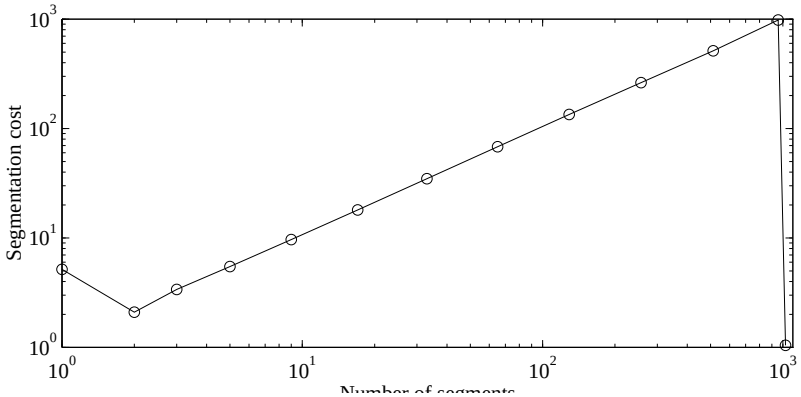


Figure 6.7: Segmentation cost versus number of segments. The minimum at 2 corresponds to the step-edge segmentation, while that at 1000 corresponds to full interpolation of the data by the reconstruction.

is varied from 0 to 1024 for a 1024-sample step. The breakpoint set for all numbers of segments greater than 2 includes the true breakpoint n. The sum-of-variance cost steadily increases with the number of breakpoints until the sample rate equals 1, at which point the model interpolates the data and the error is 0. There are therefore two local minima of the cost in terms of the number of breakpoints, corresponding to the two natural scales present in the data. Although the ability to detect natural scale is interesting, it is not further explored in this work.

Figure 6.8 shows a more compelling example—the sampled data may be considered as either a noisy parabolic arc, a sine wave superimposed on a parabolic arc, or the interpolating segmentation. Local minima of the error *as a function of number of breakpoints* are found corresponding to each of these, and to no other segmentations.

6.6 A robust variant

Modifying the metric to deal with outlier noise is the same problem as that faced by robust least squares. A robust metric must be found and an algorithm to minimize it. The simplest robust metric is given by substituting for variance the median of squared residuals [53]. Although LMS estimation is normally a computationally expensive process, for the piecewise constant case it is trivial: simply extract the median of the data. Modifying the metric to use LMS rather than variance, Figure 6.9, which is corrupted both by spatially varying and outlier noise is segmented at the correct breakpoint. (Changing the metric to the median has no effect on the segmentation produced in the previous examples, which lacked outlier noise).

6.7 Relation to previous work

The most closely related previous work is almost certainly that of Leclerc [66], who presents the combined Bayesian and information-theoretical derivation of MDL and applies it to the segmentation task. Recent work on MDL has done little to change the basic formulation, generally tending to concentrate on the nontrivial task of devising an efficient implementation. Leclerc also deals with the case of unknown spatially varying noise, and adds a $\sum_{m=1}^{M} \log_2 \sigma_m$ term to the basic formulation. The significant difference with the

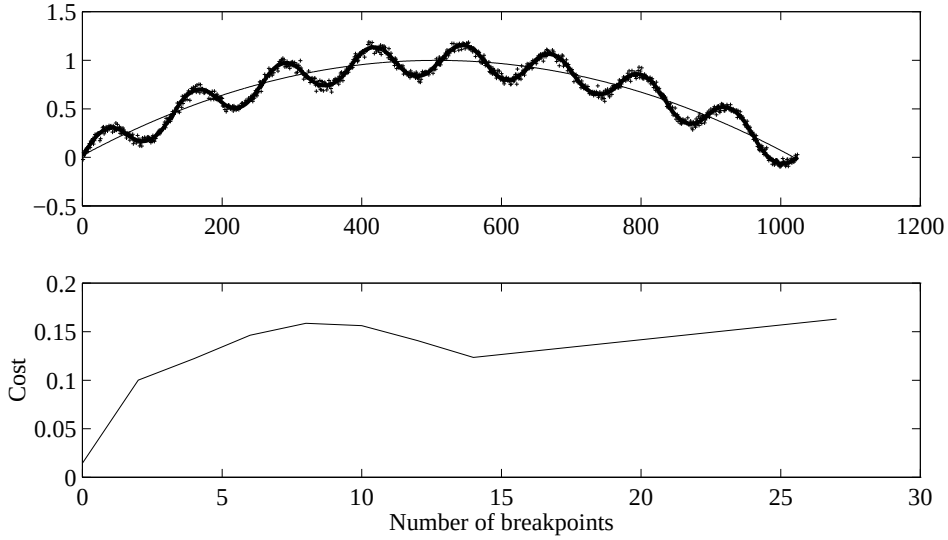


Figure 6.8: Natural scales of description. The local minima of $\sum \sigma^2$ as a function of number of breakpoints indicate the two interpretations of the data. The first minimum, at 0, corresponds to the interpretation as a parabolic arc while the second, at 15, corresponds to the approximation of the sine wave by parabolic segments. The genetic algorithm can be directed towards a desired solution that is not the global minimum by adding penalty terms, corresponding to introducing *a priori* knowledge.

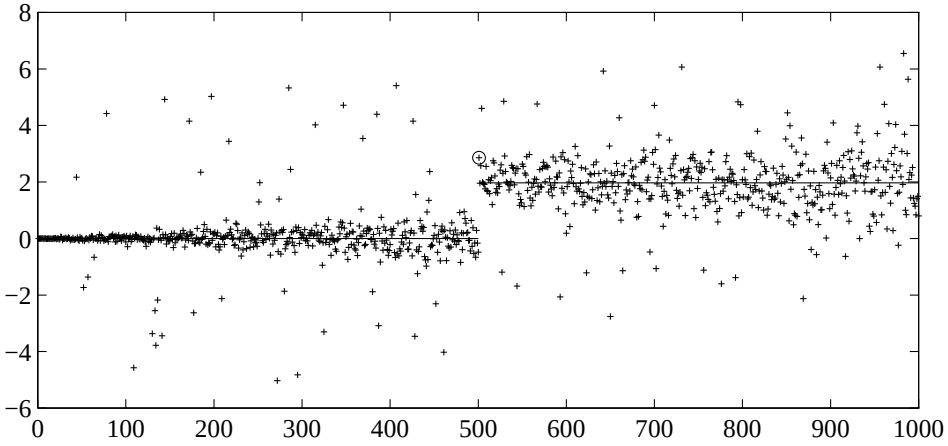


Figure 6.9: The robust variant: synthetic piecewise-constant image of a step edge of height 2, with Gaussian noise whose variance increases linearly from 0 on the left to 1 on the right. In addition, the entire edge is corrupted by 10% outlier noise. The minimum of $\sum_m \text{median}\{r_i^2 : i_m \leq i < i_{m+1}\}$ is achieved at the (correct) step-edge interpretation.

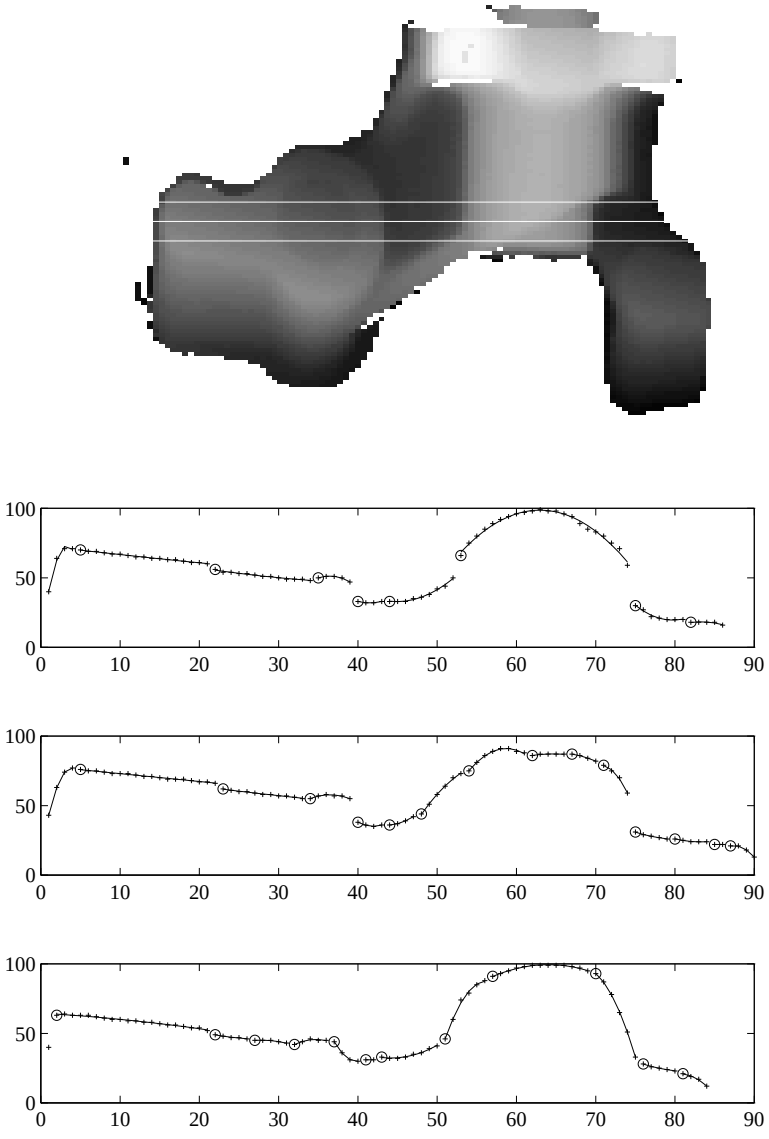
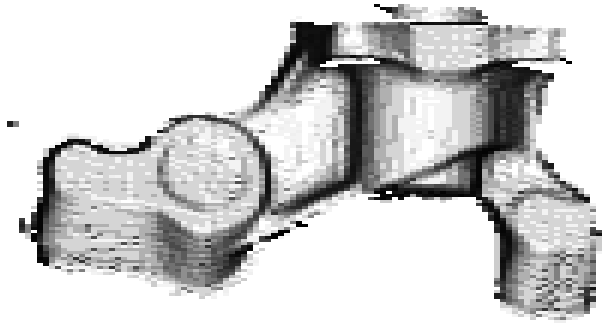


Figure 6.10: Segmentation of data from a structured light laser scanner. The curves are horizontal cross-sections of the image at the indicated positions. The segmentations into quadratic curves are shown overlaid on the data. The indentation between samples 20 and 35 has proved particularly challenging for many algorithms we have tried in the past. The sum-of-variance metric identifies it in two of the three images without the need for any MDL penalty terms or *a priori* estimated parameters. Execution time for each segment was about 10 seconds on a Sparc 10/51.

(Original, Lambertian shaded)



(SOV)

(MDL)

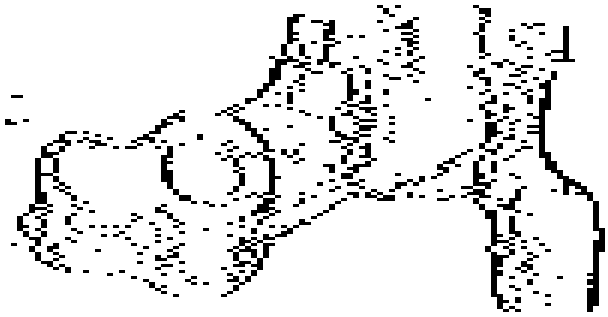


Figure 6.11: Representation of results on all stripes of the Renault image. Upper picture is sum-of-variance, lower MDL with $\lambda = 1$. Although both images are quite messy, the SOV results are more internally consistent, with thinner boundaries. The glitches on the SOV image come because in an area of the image where the variance is zero (i.e. the background) the number of breakpoints does not affect the SOV score.

sum-of-variance formulation, is that we discard the original sum of residuals that he maintains, so greatly simplifying the metric. The relationship to Lowe's segmenter, discussed in the introduction, is also of interest.

6.8 Conclusions

Figures 6.9, 6.10 and 6.11 show results of the genetic algorithm implementation on real and synthetic images. The step edge example is notable because the noise level is varying within the signal, without causing any under or oversegmentation. The Renault part image (Fig. 6.10) is a particularly challenging real example. Finally, on the multiscale example in Figure 6.8, a version of the GA biased against segmentations with fewer than 10 breakpoints models the sine wave as a sequence of parabolic arcs while the unmodified error function identifies the data as a single parabola.

The sum of variance metric (and its robust counterpart) is possibly the simplest yet proposed for image segmentation, but is arguably the most effective. The absence of user-specified thresholds renders it useful for a wide variety of tasks, and the ability to identify natural scale in the data is perceptually interesting.

It is important to note what this chapter is not: it is not presenting a new algorithm for image segmentation, it is presenting a new cost function which may be incorporated into existing algorithms – the fastest and most promising being that of Zhu [135]; it is not an "adaptive" algorithm in the sense of locally estimating the noise in parts of the image and then using the estimate to tune a standard noise-level-sensitive technique. It is a true global error that happens to work in spatially-varying noise conditions.

Future work will be centered on the adaptation of Zhu's region-growing technique to the minimization of SOV in the 2D case, which would be impractical using the GA. Finally, implementation as a self-organizing neural network has been mooted, which if successful could have important consequences for theories of human vision.

Chapter 7

Conclusion

The computer vision problem depends on the representation of shape. Representations must be powerful enough to model the interesting aspects of all images we see, yet simple enough that their extraction from image data is robust, repeatable and tractable. There is often a tradeoff between scope of representation and the stability of techniques for recovery of that representation. Current research into segmentation is centered on improving robustness while simultaneously improving the scope of systems. This thesis addresses these issues from a number of theoretical and pragmatic viewpoints, providing a useful toolkit of concepts and algorithms for vision research. The central claim of the thesis is that the tradeoff between scope of representation and stability of segmentation defines the direction in which solutions to the segmentation problem must be sought.

The exploration of this claim entailed three main bodies of work. The thesis addressed the problem of scope in the area of conic fitting by considering the restriction of general conic fitters to the ellipse-specific case, and showed that this restriction improved the stability of the algorithm. The development of the run-distribution test addressed the problem of stability with respect to *a priori* tuning parameters—the algorithmic stability referred to in the introduction. The sum-of-variance metric addressed the same issue by considering the overall problem of signal segmentation, and introduced another parameter-free metric, this time specifically for image segmentation. In combination, these techniques are a step towards the overall goal of unsupervised and stable vision systems.

7.1 Contributions

The contributions made by this thesis to the state of the art of computer vision research include:

- A new direct method for the least-squares fitting of ellipses. Previous methods have been computationally expensive or have not been ellipse-specific, often returning arbitrary conics. The extension of this method to the other specific case of hyperbolae.
- The introduction of the *run-distribution* test, a method of statistical goodness-of-fit testing which, being distribution-free, is superior to the existing methods commonly used in computer vision in the common case that the data noise distribution is poorly known. The extension of this method to the case where the form of the noise distribution is known, although its parameters are not.
- The introduction of the sum-of-variance metric, a metric for goodness of fit which applies to piecewise smooth curves and surfaces. The metric is again independent of the data noise levels.

7.2 Conclusions

The conclusions which I draw from the work presented in this thesis are drawn first for each contribution individually and then in concert.

7.2.1 Conic fitting

The ellipse-specific fitter is beneficial for two reasons: first, the problem of how to consistently fit ellipses under high occlusion has been approached a number of times over the last twenty years, and the solutions proposed have all been computationally expensive; second, because the scope of shape representation is less than that of the general conic fitters, our algorithm shows significantly greater robustness to noise (Figure 3.8) and to occlusion than the general algorithms. However, it has the disadvantage that it suffers the high curvature bias of many conic fitting techniques, which means that the estimated conic parameters are less accurate than those which may be obtained by the iterative methods. Given the requirements of speed and stability, particularly in situations of high noise, it performs well, but in more typical low-noise situations should be used as a backup for the Taubin algorithm.

The evaluation of conic fitting is a useful complement to the work of [104, 106], providing an empirical survey of a number of conic fitters. Although the survey introduces no new theoretical results, the methodology used addresses the difficult problems of how to compare the algorithms and how to efficiently but comprehensively explore the space of possible input data and therefore to present results which can be extrapolated to a wider range of problems. The principal criticism of the study is the range of algorithms covered. By concentrating on the faster "direct" algorithms, it may be argued that the results are less useful to practitioners whose primary interest is in accuracy rather than speed of execution. Countering this, the survey does include the algorithms (geometric distance and Kanatani's) which are theoretically the most accurate and finds that at least one of the direct approaches (the Taubin method) is very similar in terms of accuracy. This provides an accuracy benchmark against which slower algorithms should be tested: if a slower algorithm is less accurate than Taubin's it provides little benefit to the practitioner, while one which is faster and more accurate will become the method of choice.

7.2.2 Hypothesis testing and segmentation

The introduction of the run-distribution test has significant relevance for the construction of automatic vision systems. Because of the relaxation of the typical requirement for an *a priori* noise estimate, systems can be built which operate "out of the box" on a variety of different data sets, without tuning the system parameters. In addition, if the characteristics of the data vary over time—for example a system which must work in varying lighting conditions—a system need not be continually recalibrated in order to provide a consistent response.

The notable characteristic of the sum-of-variance metric demonstrated here is similar: its ability to reproduce the ground-truth segmentations of noisy data, without any estimate of the noise level. Its ability to achieve this in the difficult cases of spatially varying noise, and on data which exhibits only curvature discontinuities indicates that it is successfully dividing the data into model and noise processes. It has been demonstrated that the results extend to real data in the 1D case. The major problem is that it has almost no tuning parameters—if it returns a segmentation that is deemed inappropriate to the task at hand, there is little opportunity to direct it towards the desired answer.

In contrast to many existing systems, which are thrust into a niche by the requirement that they be carefully tuned and placed in carefully engineered environments, a system based on the principles embodied in the run-distribution and sum-of-variance metrics has the capability to be of generic applicability. Systems designed on such parameter free

principles can be freed from the domain of well-engineered and controlled environments, and point towards a more generic visual competence.

On the other hand, if a system is to be installed in a controlled situation, where the environmental influences remain largely static over time, superior performance will be achieved by using metrics which have been tuned for the particular application and environment at hand. Using the run-distribution test for example: although its performance remains constant over a wide class of input data, it is outperformed when competing in the relatively narrow niche occupied by the correctly tuned chi-squared test. Figure 5.10 demonstrates that RD segmentation operates at a level equivalent to MDL when the latter is mistuned by about 20%, and equals a correctly tuned chi-squared test under high noise.

This has important implications for the design of generic vision systems: it may be necessary to relinquish the ability to engineer the environment in favour of more clever engineering of the system. A system which can use run-distribution or sum-of-variance output to achieve its design objectives—even though that output may be inferior to that available in a restricted environment—will be freed from reliance on an accurate noise model and immediately gain genericity.

7.2.3 General conclusions

The claim of the introduction that scope and stability must of necessity imply a tradeoff is interestingly demonstrated in two rather different ways by the work reported here. The more obvious demonstration is in the increased stability of the ellipse-specific fitter in Chapter 3 and the circle-specific fitters in Chapter 4. In this case the restriction of scope provides a clear increase in robustness to occlusion, and therefore to stability.

The second demonstration is in some ways more surprising: the parameterless error metrics in Chapters 5 and 6 *extend* the scope of a system by allowing it to work on a broader range of input data, rather than being restricted to a specific noise model and level. This in turn leads to a reduction in the sensitivity of the run-distribution test, and a reduction in applicability of the sum of variance metric. Although the extent of this reduction is relatively small, it is significant. Because the goal of the work was to explore the performance of the "pure" metrics, with the minimum of hidden parameters, the actual position taken along the scope-stability tradeoff curve was very much biased towards wide scope. When the incorporation of the techniques into a real system is considered, the genericity of the domain of application will determine the extent to which these techniques are adopted: A system which wishes to operate in a generic domain must therefore be prepared to accept the concomitant reduction in stability, and the search for such a generic vision system must have, as a primary goal, the objective of optimizing this tradeoff in order to simultaneously maximize scope and stability.

7.3 Further work

The primary area for extension of these results is to address the scope-stability tradeoff and design directly to it. Preliminary work in this direction was presented in [35], where a new curve representation, the monotonically deformable parabola was introduced as a means of extending the scope of the simplest of curve representations while maintaining the property that the curve has a single extremum. Although the research was inconclusive—the expense of the fitting algorithm dictating that meaningful experiments were not performed—its objective of directly addressing the scope-stability tradeoff is in the spirit of the conclusions of the thesis and therefore benefit might accrue from addressing the speed issues and exploring this work further.

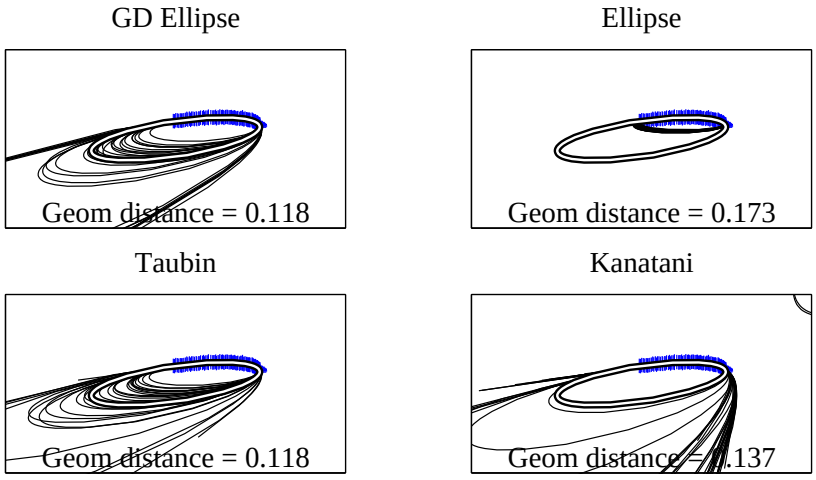


Figure 7.1: Bias corrected ellipse-specific fitting. The ellipse center is moved along the line joining the fitted center and the midpoint of the dataset. This generates a one-parameter family of ellipses, and this 1D function is optimized to find the minimum geometric distance. Speed is about half of Kanatani, accuracy about that of Taubin, but no hyperbolae may be produced.

7.3.1 Conic fitting

In the area of conic fitting, the most important and valuable extension will come from applying the Kanatani and Porrill bias-correction technique to the ellipse specific fitter. Chapter 4, and specifically Figure 4.9, indicate that the true center position is always on the line joining from the midpoint of the data set to the reported center, and furthermore that this bias is not a function of position around the arc of the data midpoint. Using this observation it is hoped that a "one-shot" bias corrected ellipse fitter can be produced, where the reported parameters and the estimated subtended angle will immediately yield the true parameters. Preliminary experiments on the validity of the prediction use an iterative method which minimizes the geometric distance (see Figure 7.1) by moving the center along the line show performance similar to Taubin's method but without the possibility of hyperbolic solutions. However the mathematics required to produce the one-shot algorithm have so far proved elusive.

7.3.2 Segmentation

In the segmentation topics, the avenues for exploration are improvement of the segmentation strategies and extension to higher dimensions.

In the case of the run-distribution/MDL comparison, a deficiency is that the simple form of the segmentation algorithm makes it more difficult to be certain that the relative merits of the tests will not change if a more sophisticated algorithm is used. Improving the algorithm may affect the relative merits of the systems. In this way, the effectiveness of the segmentation strategy need not be taken into account when comparing the metrics. This would allow a deeper understanding of the relationships between the alternative techniques, and might lead to a hybrid whose tunabilty is itself tunable—allowing the user of the techniques to select the position along the scope-stability curve which is most appropriate for the specific application under development. For the sum-of-variance metric, the problem is reversed—the genetic algorithm essentially ensures that the optimal segmentation has been achieved, but renders the algorithm too slow to test in 2D. Work on a more heuristic or problem-specific approach such as region growing or minimization over multiple scales is

a possible strategy.

In both cases, the extension of the methods to higher dimensions represents an important increase in applicability. These extensions are however rather different. In the case of the sum-of-variance metric, the metric itself extends trivially to any dimension, but the algorithm for its minimization becomes exponentially more expensive as dimensionality is increased. In the case of the run-distribution segmenter, the difficulty lies in extending the definition of the metric itself to higher dimensions. Although a possible method is presented in Chapter 5, it is clear that significant work is required in order to evaluate the extension.

Appendix A

Proof of Lemma 1

This appendix details the proof of lemma 1. It is a variation on the proof of Pilu [88]. The general reference followed is [43].

Lemma 1 *The signs of the generalized eigenvalues of*

$$\mathbf{S}\mathbf{u} = \lambda\mathbf{C}\mathbf{u} \tag{A.1}$$

where $\mathbf{S} \in \Re_{n\times n}$ *is positive definite, and* $\mathbf{C} \in \Re_{n\times n}$ *is symmetric, are the same as those of the matrix* $\mathbf{C}$*, up to permutation of the indices.*

Let us define the *spectrum* $\sigma(\mathbf{S})$ as the set of eigenvalues of $\mathbf{S}$. Let us analogously define $\sigma(\mathbf{S},\mathbf{C})$ to be the set of generalized eigenvalues of (A.1). The *inertia* $i(\mathbf{S})$ is defined as the set of signs of $\sigma(\mathbf{C})$, and we correspondingly define $i(\mathbf{S},\mathbf{C})$. Then the lemma is equivalent to proving that $i(\mathbf{S},\mathbf{C}) = i(\mathbf{C})$.

As $\mathbf{S}$ is positive definite, it may be decomposed as $\mathbf{Q}^2$ for symmetric $\mathbf{Q}$, allowing us to write (A.1) as

$$\mathbf{Q}^2\mathbf{u} = \lambda\mathbf{C}\mathbf{u}$$

Now, substituting $\mathbf{v} = \mathbf{Q}\mathbf{u}$ and premultiplying by $\mathbf{Q}^{-1}$ gives

$$\mathbf{v} = \lambda\mathbf{Q}^{-1}\mathbf{C}\mathbf{Q}^{-1}\mathbf{v}$$

so that $\sigma(\mathbf{S},\mathbf{C}) = \sigma(\mathbf{Q}^{-1}\mathbf{C}\mathbf{Q}^{-1})^{-1}$ and thus $i(\mathbf{S},\mathbf{C}) = i(\mathbf{Q}^{-1}\mathbf{C}\mathbf{Q}^{-1})$.

From Sylvester's Law of Inertia [132] we have that for any symmetric $\mathbf{S}$ and nonsingular $\mathbf{X}$,

$$i(\mathbf{S}) = i(\mathbf{X}^T\mathbf{S}\mathbf{X})$$

Therefore, substituting $\mathbf{X} = \mathbf{X}^T = \mathbf{Q}^{-1}$ we have $i(\mathbf{C}) = i(\mathbf{Q}^{-1}\mathbf{C}\mathbf{Q}^{-1}) = i(\mathbf{S},\mathbf{C})$. □

Appendix B

Conic section geometry

For completeness, the precise formulae used for extraction of principal points and radii from conics are given here. The conic is

$$ax^2 + bxy + cy^2 + dx + ey + f = 0,$$

And we extract

$$\theta = \frac{1}{2}\tan^{-1}\frac{a-c}{b}$$

Rotating coordinates so the conic is

$$A_{uu}u^2 + A_{vv}v^2 + A_u u + A_v v + A_o = 0,$$

We have

$$\begin{aligned} A_o &= f \\ A_u &= d\cos(\theta) + e\sin(\theta) \\ A_v &= -d\sin(\theta) + e\cos(\theta) \\ A_{uu} &= a\cos^2(\theta) + c\sin^2(\theta) + b\cos(\theta)\sin(\theta) \\ A_{vv} &= a\sin^2(\theta) + c\cos^2(\theta) - b\cos(\theta)\sin(\theta) \end{aligned}$$

Then the centre or principal point in rotated coordinates is

$$\begin{aligned} u_0 &= -\frac{A_u}{2A_{uu}} \\ v_0 &= -\frac{A_v}{2A_{vv}} \end{aligned}$$

which is transformed back to

$$\begin{aligned} c_x &= u_0\cos(\theta) - v_0\sin(\theta) \\ c_y &= u_0\sin(\theta) + v_0\cos(\theta) \end{aligned}$$

And the radii are given by

$$k = A_o - A_{uu}u_0^2 - A_{vv}v_0^2$$

$$R_x = -\sqrt{\frac{k}{A_{uu}}}$$

$$R_y = -\sqrt{\frac{k}{A_{vv}}}$$

where negation is passed through the square root. The original conic is now of the form

$$\frac{(x - c_x)^2}{R_x|R_x|} + \frac{(y - c_y)^2}{R_y|R_y|} = 1$$

And the principal points are at $(c_x \pm R_x, c_y \pm R_y)$.

Appendix C

Invariance Properties of Conic Fitting Methods

In order to demonstrate the invariance of the conic fitting algorithms to transformations of the data points, we follow the method of [38]. First we define an invariant of a parameterization:

> Given a family of curves parameterized by the vector $\mathbf{a}$, and a group of transformations $\mathcal{G}$, a scalar function $I(\mathbf{a})$ of the elements of $\mathbf{a}$ is an *invariant of weight* ω under $\mathcal{G}$ if
>
> $$I(\mathbf{a}') = I(\mathbf{a})|\mathbf{g}|^{\omega}$$
>
> where $\mathbf{g}$ is an arbitrary member of $\mathcal{G}$ and $\mathbf{a}'$ is the new parameter vector that represents the same curve as $\mathbf{a}$ but in the new coordinate system given by the transformation $\mathbf{g}$. The determinant of a group member $|\mathbf{g}|$ is defined as the determinant of its matrix representation if it is a linear function, which is all we will deal with here.

Forsyth *et al.* then show that fitting subject to the constraint $I(\mathbf{a}) =$ constant is invariant to transformations from a group $\mathcal{G}$ precisely when I is an invariant (of any weight) under $\mathcal{G}$.

While their proof is somewhat involved, a simpler version may be derived for the cases of interest in the current discussion. Specifically, we may use the fact that minimization of

$$\mathbf{a}^{\top} S \mathbf{a} \quad \text{subject to } I(\mathbf{a}) = 1$$

where $I(\mathbf{a})$ is of the special form $\mathbf{a}^{\top} C \mathbf{a}$ is equivalent to the minimization of the Rayleigh quotient:

$$R = \frac{\mathbf{a}^{\top} S \mathbf{a}}{I(\mathbf{a})}$$

Then the error metric in the transformed frame

$$R' = \frac{\mathbf{a}'^{\top} S' \mathbf{a}'}{I(\mathbf{a}')}$$

is a scalar multiple of the error metric in the original frame (specifically it is divided by $|\mathbf{g}|^{\omega}$). Moreover this scalar multiple is independent of $\mathbf{a}$ so that the metric is scaled equally for all vectors $\mathbf{a}$—no particular $\mathbf{a}$ is favoured over any other once $\mathbf{g}$ acts on them all equally. Therefore the minimizer in the transformed frame will be the transform of the minimizer in the original frame.

C.1 The affine group

A member of the group of affine transforms may be represented by its action on points in the plane using the matrix-vector equation

$$\mathbf{x}' = M\mathbf{x} + \mathbf{t}$$

where M is an invertible 2×2 matrix, with $\mathbf{t}$ a 2D translation vector. The determinant of such a transform is

$$\begin{vmatrix} M & \mathbf{t} \\ \mathbf{0}_{1\times 2} & 1 \end{vmatrix}$$

Expanding this determinant by its bottom row, we see that it evaluates simply to $|M|$, with the translation component playing no part.

C.2 Matrix representation of the conic

Let us represent the conic in the matrix form

$$\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{x}^T\mathbf{b} + f = 0 \tag{C.1}$$

where the matrices are related to the implicit parameters in Chapter 2, (equation (2.1)) by:

$$\mathbf{Q} = \begin{pmatrix} a & b/2 \\ b/2 & c \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} d \\ e \end{pmatrix}$$

If we apply a change of coordinates $\mathbf{x} \to \mathbf{x}'$, the conic represented in the new coordinate system is

$$\mathbf{x}'^T\mathbf{Q}'\mathbf{x}' + \mathbf{x}'^T\mathbf{b}' + f' = 0$$

In order to relate the primed parameters to the original ones, we expand in terms of the original points $\mathbf{x}$:

$$(M\mathbf{x} + \mathbf{t})^T\mathbf{Q}'(M\mathbf{x} + \mathbf{t}) + (M\mathbf{x} + \mathbf{t})^T\mathbf{b}' + f' = 0.$$

Gathering terms in $\mathbf{x}$ and equating with (C.1), we get

$$\begin{aligned} \mathbf{Q} &= M^T\mathbf{Q}'M, \\ \mathbf{b} &= M^T(2\mathbf{Q}'\mathbf{t} + \mathbf{b}'), \\ f &= \mathbf{t}^T\mathbf{Q}'\mathbf{t} + \mathbf{t}^T\mathbf{b}' + f'. \end{aligned}$$

Inversion of these equations yields expressions for the primed parameters in terms of the originals:

$$\begin{aligned} \mathbf{Q}' &= M^{-T}\mathbf{Q}M^{-1}, \\ \mathbf{b}' &= M^{-T}\mathbf{b} - 2\mathbf{Q}'\mathbf{t} \\ &= M^{-T}\mathbf{b} - 2M^{-T}\mathbf{Q}M^{-1}\mathbf{t}, \\ f' &= \mathbf{t}^T\mathbf{Q}'\mathbf{t} - \mathbf{t}^TM^{-T}\mathbf{b}' + f. \end{aligned}$$

C.3 Invariance of the discriminant

Then, noting that the constraint $4ac - b^2 = 1$ is equivalent to fixing the determinant $|\mathbf{Q}| = \frac{1}{4}$, we have that

$$\begin{aligned} |\mathbf{Q}'| &= |M^{-T}\mathbf{Q}M^{-1}| \\ &= |M^{-T}||\mathbf{Q}||M^{-1}| \\ &= |\mathbf{Q}||M|^{-2} \end{aligned}$$

and therefore that the discriminant is an affine invariant of weight -2. Applying the result of Forsyth *et al.* shows that fitting with prescribed $|\mathbf{Q}|$ is invariant to affine transformations of the data □

C.4 Similarity invariance of Bookstein's constraint

It is instructive also to demonstrate the invariance of the Bookstein constraint to similarity transformations using the matrix formulation. The constraint $a^2 + \frac{b^2}{2} + c^2$ is expressed in the matrix form as

$$\text{trace}(\mathbf{Q}^2)$$

Under the action of a similarity transformation $\mathbf{g}(\mathbf{x}) = sR\mathbf{x} + \mathbf{t}$, where R is of determinant 1 and s is a scalar, this transforms to (noting that $R^{-1} = R^T$):

$$\begin{aligned}
\text{trace}(\mathbf{Q}'^2) &= \text{trace}((s^{-2}R\mathbf{Q}R^T)^2) \\
&= s^{-4}\text{trace}(R\mathbf{Q}R^T R\mathbf{Q}R^T) \\
&= s^{-4}\text{trace}(R\mathbf{Q}\mathbf{I}\mathbf{Q}R^T) \\
&= s^{-4}\text{trace}(\mathbf{Q}^2) \quad (\text{using } |R| = 1) \\
&= |\mathbf{g}|^{-2}\text{trace}(\mathbf{Q}^2)
\end{aligned}$$

Which is again an invariant of weight -2 □

C.5 Similarity invariance of Taubin's constraint

Taubin's constraint is interesting in that it depends both on the conic parameters and on the values of the data points. In this section, I demonstrate that it is invariant to similarity transformations. The constraint is of the form

$$\sum_{i=1}^{n} \|2\mathbf{Q}\mathbf{x}_i + \mathbf{b}\|^2$$

and we shall first examine the behaviour of $2\mathbf{Q}\mathbf{x} + \mathbf{b}$ under affine transformation $\mathbf{x}' = M\mathbf{x} + \mathbf{t}$.

$$\begin{aligned}
2\mathbf{Q}'\mathbf{x}' + \mathbf{b}' &= 2\mathbf{Q}'M\mathbf{x} + 2\mathbf{Q}'\mathbf{t} + M^{-T}\mathbf{b} - 2\mathbf{Q}'\mathbf{t} \\
&= 2\mathbf{Q}'M\mathbf{x} + M^{-T}\mathbf{b} \\
&= 2M^{-T}\mathbf{Q}M^{-1}M\mathbf{x} + M^{-T}\mathbf{b} \\
&= M^{-T}(2\mathbf{Q}\mathbf{x} + \mathbf{b})
\end{aligned}$$

Geometrically, this has the simple interpretation that the normal to the curve is transformed by the inverse of the transformation applied to the data points. Now, in the case of similarity transformations $\mathbf{x}' = sR\mathbf{x} + \mathbf{t}$, the norm

$$\begin{aligned}
\|2\mathbf{Q}'\mathbf{x}' + \mathbf{b}'\|^2 &= (2\mathbf{Q}\mathbf{x} + \mathbf{b})^T s^{-2} RR^T (2\mathbf{Q}\mathbf{x} + \mathbf{b}) \\
&= s^{-2}\|2\mathbf{Q}\mathbf{x} + \mathbf{b}\|^2
\end{aligned}$$

and the constraint is again an invariant of weight -2 □.

Bibliography

[1] H. Akaike. A new look at statistical model identification. *IEEE Transactions on Automatic Control*, 19:716, 1974.

[2] A. Albano. Representation of digitized contours in terms of conic arcs and straight-line segments. *Computer Graphics and Image Processing*, 3:23–33, 1975.

[3] A. Asada and M. Brady. The curvature primal sketch. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 8(1):2–14, January 1986.

[4] N. J. Ayache and O. D. Faugeras. Hyper: A new approach for the recognition and positioning of two-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 8(1):44–54, 1986.

[5] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[6] A.H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23, 1981.

[7] Samuel W. Bent and John W. John. Finding the median requires $2n$ comparisons. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, pages 213–216, Providence, Rhode Island, 6–8 May 1985.

[8] P. J. Besl and R. C. Jain. Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 10(2):167–192, March 1988.

[9] R. H. Biggerstaff. Three variations in dental arch form estimated by a quadratic equation. *Dental Research*, 51:1509–1516, 1972.

[10] A. Blake, R. Curwen, and A. Zisserman. Affine-invariant contour tracking with automatic control of spatiotemporal scale. In *Proceedings, International Conference on Computer Vision*, pages 66–75, 1993.

[11] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, London, 1987.

[12] R. M. Bolle and B. C. Vemuri. On three-dimensional surface reconstruction methods. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 13(1):1–13, 1991.

[13] R. C. Bolles and R. A. Cain. Recognizing and locating partially visible objects: The local feature focus method. *International Journal of Robotics Research*, 1(3):637–643, 1982.

[14] R. C. Bolles and M. A. Fischler. A RANSAC-based approach to model fitting and its application to finding cylinders in range data. In *Proceedings, International Joint Conference on AI*, pages 637–643, 1981.

[15] R. C. Bolles and P. Horaud. 3DPO: A three-dimensional part orientation system. *International Journal of Robotics Research*, 5(Fall):3–26, 1986.

[16] F. Bookstein. Fitting conic sections to scattered data. *Computer Graphics and Image Processing*, 9:56–71, 1979.

[17] J. M. Brady. Criteria for representations of shape. In *Human and Machine Vision*. Academic Press, 1983.

[18] R. A. Brooks. Symbolic reasoning among 3D models and 2D images. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 5(2):140–149, March 1983.

[19] K.A. Brownlee. *Statistical Theory and Methodology in Science and Engineering*. Wiley, 1960.

[20] L. D. Cai. A diffusion smoothing approach to sculptured surfaces. In *The Mathematics of Surfaces III*, pages 267–281. Clarendon Press, Oxford, 1989.

[21] J. Canny. A computational approach to edge-detection. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 8(6):679–698, 1986.

[22] J. F. Canny. Finding edges and lines in images. Technical Report 720, MIT AI Lab, 1983.

[23] S. Carlsson. Projectively invariant decomposition and recognition of planar shapes. *International Journal of Computer Vision*, 17(2):193–209, 1996.

[24] D. S. Chen. A data-driven intermediate level feature extraction algorithm. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 11(7):749–758, July 1989.

[25] D. R. Cooper and N. Yalabik. On the computational cost of approximating and recognizing noise-perturbed straight lines and quadratic arcs in the plane. *IEEE Transactions on Computers*, 25:1020ff, 1976.

[26] K. Daniilidis and J. Ernst. Active intrinsic calibration using vanishing points. *Pattern Recognition Letters*, 17(11):1179–1189, 1996.

[27] T. Darrell, S. Sclaroff, and A. Pentland. Segmentation by minimal description. In *Proceedings, International Conference on Computer Vision*, 1990.

[28] C. de Boor. *A practical guide to splines*. Springer-Verlag, New York, 1978.

[29] J. G. Dunham. Optimum uniform piecewise linear approximation of planar curves. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 8(1):67–75, January 1986.

[30] T. Ellis, A. Abbood, and B. Brillault. Ellipse detection and matching with uncertainty. *Image and Vision Computing*, 10(2):271–276, 1992.

[31] T.-J. Fan. *Describing and recognizing 3-D objects using surface properties*. Springer-Verlag, 1990.

[32] O. Faugeras and M. Hebert. A 3-d recognition and positioning algorithm using geometric matching between primitive surfaces. *International Journal of Robotics Research*, 5(3):27–52, 1986.

[33] R. B. Fisher. Model invocation for three dimensional scene understanding. In *Proceedings, International Joint Conference on AI*, pages 805–807, 1987.

[34] R. B. Fisher. *From Surfaces to Objects: Computer Vision and Three Dimensional Scene Analysis*. John Wiley, UK, 1989.

[35] A. W. Fitzgibbon and R. B. Fisher. Invariant fitting of arbitrary single-extremum surfaces. In *Proceedings, British Machine Vision Conference*, pages 569–578, September 1993.

[36] A. W. Fitzgibbon and R. B. Fisher. A buyer's guide to conic fitting. In *Proceedings, British Machine Vision Conference*, pages 513–522, 1995.

[37] P. Flynn. *CAD-Based Computer Vision: Modelling and Recognition Strategies*. PhD thesis, Michigan State University, 1990.

[38] D. Forsyth, J. L. Mundy, A. Zisserman, and C. M. Brown. Projectively invariant representations using implicit algebraic curves. *Image and Vision Computing*, 9(2):130–136, April 1991.

[39] D. Forsyth, J. L. Mundy, A. Zisserman, et al. Invariant descriptors for 3D object recognition and pose. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 13(10):971–991, 1991.

[40] W. Gander, G.H. Golub, and R. Strebel. Least-square fitting of circles and ellipses. *BIT*, 43:558–578, 1994.

[41] M. Gardiner. The superellipse: a curve that lies between the ellipse and the rectangle. *Scientific American*, September 1965.

[42] R. Gnanadesikan. *Methods for statistical data analysis of multivariate observations*. John Wiley & sons, New York, 1977.

[43] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins, 2^{nd} edition, 1989.

[44] W. E. L. Grimson and T. Lozano Pérez. Model-based recognition and localization from sparse range or tactile data. *International Journal of Robotics Research*, 3(3), 1984.

[45] W.E.L. Grimson. *Object recognition by computer : the role of geometric constraints*. MIT Press, 1990.

[46] A. Gupta and R. S. Bajcsy. Volumetric segmentation of range images of 3D objects using superquadric models. *CVGIP: Image Understanding*, 58(3):302–326, 1993.

[47] A. Hanson and E. Riseman. The VISIONS image-understanding system. In C. Brown, editor, *Advances in Computer Vision*. Lawrence Erlbaum Assoc, NJ 07642, 1988.

[48] R. Haralick. Performance characterization in computer vision. In *Proceedings, British Machine Vision Conference*, pages 1–8, 1992.

[49] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*, volume 1. Addison-Wesley, 1993.

[50] R. I. Hartley. In defence of the 8-point algorithm. In *Proceedings, International Conference on Computer Vision*, pages 1064–1070, 1995.

[51] C. M. Hoffmann. Implicit curves and surfaces in CAGD. *IEEE Computer Graphics and Applications*, 13(1):79–88, 1993.

[52] P. V. C. Hough. Method and means for recognising complex patterns., Dec 1962. U.S. Patent No. 3,069,654.

[53] P.J. Huber. *Robust Statistics*. Wiley, New York, 1981.

[54] A. K. Jain and R. Hoffman. Evidence-based recognition of 3-D objects. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 10(6):783ff, November 1988.

[55] K. Kanatani. Statistical bias of conic fitting and renormalization. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 16(3):320–326, 1994.

[56] K. Kanatani and W. Liu. 3-D interpretation of conics and orthogonality. *CVGIP: Image Understanding*, 58(3):286–301, 1993.

[57] D. Kapur and J. L. Mundy. Fitting affine-invariant conics to curves. In J. L. Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*, page (). MIT Press, 1992.

[58] M. Kass, A. Witkin, and D. Terzopulos. Snakes: active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.

[59] O. Kempthorne and L. Folks. *Probability, Statistics, and Data Analysis*. Iowa State University Press, Ames, Iowa 50010, 1971.

[60] M. Kendall and A. Stuart. *The Advanced Theory of Statistics*, volume 2. Charles Griffin and Company, London, 3rd edition, 1973.

[61] D. Keren, D. Cooper, and J. Subrahmonia. Describing complicated objects by implicit polynomials. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 16(1):38–53, 1994.

[62] J. Kittler. Tutorial on statistics in machine vision. In *Proceedings, British Machine Vision Conference*, 1993.

[63] A. J. Lacey, N. A. Thacker, and N. L. Seed. Feature tracking and motion classification using a switchable model Kalman filter. In *Proceedings, British Machine Vision Conference*, pages 599–608, 94.

[64] Y. Lamdan and H. J. Wolfson. On the error analysis of "geometric hashing". In *IEEE Conf. on Computer Vision and Pattern Recognition*, 1991.

[65] V.F. Leavers. *Shape Detection in Computer Vision Using the Hough Transform*. Springer-Verlag, 1992.

[66] Y.G. Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 3:73–102, 1989.

[67] D. Lee and T. Pavlidis. One dimensional regularisation with discontinuities. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 10(6):822–829, 1988.

[68] A. Leonardis, F. Solina, and A. Macerl. A direct recovery of superquadric models in range images using recover-and-select paradigm. In *ECCV*, pages 309–318. Springer-Verlag, 1994.

[69] Ales Leonardis. *Image Analysis using parametric models*. PhD Thesis, University of Ljubljana, Slovenia, 1993.

[70] M. Li. Minimum description length based 2-D shape description. In *Proceedings, International Conference on Computer Vision*, 1993.

[71] T. Lindeberg and M. Li. Automatic generation of break points for MDL based curve classification. Technical report, ISRN KTH Stockholm, 1995.

[72] D. G. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, 1985.

[73] D. Marr. *Vision*. W. H. Freeman Company, New York, 1982.

[74] D. Marr and H. K. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London, Series B*, 200:269–294, 1978.

[75] The MathWorks, Inc, Natick MA, USA. MATLAB *Reference Guide*, 1992.

[76] P. Meer, D. Mintz, A. Rosenfeld, and D. Y. Kim. Robust regression methods for computer vision: A review. *International Journal of Computer Vision*, 6(1):59–70, 1991.

[77] F. Mokhtarian and A. Mackworth. Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 8(1):34–43, January 1986.

[78] A.M. Mood. The distribution theory of runs. *Ann. Math. Stat.*, 14:217–226, 1940.

[79] J. L. Mundy. The Image Understanding Environment program. *IEEE Expert—Intelligent Systems and their Applications*, 10(6):64–73, 1995. See http://www.aai.com.

[80] Y. Nakagawa and A. Rosenfeld. A note on polygonal and elliptical approximation of mechanical parts. *Pattern Recognition*, 11:133–142, 1979.

[81] M. Oshima and Y. Shirai. Object recognition using three-dimensional information. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 5(4):353–361, 1983.

[82] K. A. Paton. Conic sections in automatic chromosome analysis. In B. Meltzer and D. Michie, editors, *Machine Intelligence 5*, pages 411–434. Edinburgh University Press, Edinburgh, 1969.

[83] T. Pavlidis. Curve fitting with conic splines. *ACM Transactions on Graphics*, 2(1):1–31, 1983.

[84] T. Pavlidis. Applications of splines to shape description. In C. Arcelli, L. P. Cordella, and G. Sanniti di Baja, editors, *Visual Form*, pages 431–441. Plenum Publishing Corp, NY 10013, 1992.

[85] E.P.D. Pednault. Some experiments in applying inductive inference principles to surface reconstruction. In *Proceedings, International Joint Conference on AI*, pages 1603–1609, Detroit, MI, August 1989.

[86] A. P. Pentland. Perceptual organisation and the representation of natural form. *Artificial Intelligence*, 28:293–331, 1986.

[87] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 12(7):629–639, 1990.

[88] M. Pilu. *Part-based Grouping and Recognition: A Model-Guided Approach*. PhD thesis, University of Edinburgh, 1996.

[89] M. Pilu and R. B. Fisher. Part segmentation from 2D edge images by the MDL criterion. In *Proceedings, British Machine Vision Conference*, pages 83–92, 1996.

[90] M. Pilu, A. W. Fitzgibbon, and R. B. Fisher. Training PDMs on models: The case of deformable superellipses. In *Proceedings, British Machine Vision Conference*, pages 373–382, 1996.

[91] J. Porrill. Fitting ellipses and predicting confidence envelopes using a bias corrected Kalman filter. *Image and Vision Computing*, 8(1):37–41, February 1990.

[92] V. Pratt. Direct least-squares fitting of algebraic surfaces. *ACM Computer Graphics*, 6(4):145–152, 1987. Proceedings of SIGGRAPH.

[93] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, second edition, 1992.

[94] L. Quan. Invariant of a pair of non-coplanar conics in space: Definition, geometric interpretation and computation. In *Proceedings, International Conference on Computer Vision*, pages 926–931, 1995.

[95] L. Quan. Conic reconstruction and correspondence from 2 views. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 18(2):151–160, 1996.

[96] L. Quan, P. Gros, and R. Mohr. Invariants of a pair of conics revisited. *Image and Vision Computing*, 10:319–323, 1993.

[97] N. S. Raja and A. K. Jain. Obtaining generic parts from range images using a multiview representation. *CVGIP: Image Understanding*, 60(1):44–64, 1994.

[98] U. Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1, 1972.

[99] S. S. Rao. *Optimization: Theory and Applications*. Wiley Eastern, 2 edition, 1984.

[100] B.K. Ray and K.S. Ray. A new split-and-merge technique for polygonal-approximation of chain coded curves. *Pattern Recognition Letters*, 16(2):161–169, February 1995.

[101] J. Rissnanen. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 2:416–431, 1983.

[102] L. G. Roberts. Machine perception of three-dimensional solids. In J. Tippett et al., editors, *Optical and Electro-Optical Information Processing*, pages 159–197. MIT Press, 1965.

[103] P. L. Rosin. Ellipse fitting by accumulating five-point fits. *Pattern Recognition Letters*, 14:661–669, 1993.

[104] P. L. Rosin. A note on the least-squares fitting of ellipses. *Pattern Recognition Letters*, 14:799–808, 1993.

[105] P. L. Rosin. Analysing error of fit functions for ellipses. In *Proceedings, British Machine Vision Conference*, pages 445–454, 1996.

[106] P. L. Rosin. Assessing error of fit functions for ellipses. *CVGIP: Graphical Models and Image Processing*, 58(5):494–502, 1996.

[107] P. L. Rosin. Techniques for assessing polygonal approximations of curves. In *Proceedings, British Machine Vision Conference*, pages 153–162, 1996.

[108] P. L. Rosin and G. A. West. Segmenting curves into lines and arcs. In *Proceedings of the Third International Conference on Computer Vision*, pages 74–78, Osaka, Japan, December 1990.

[109] P. L. Rosin and G. A. W. West. Nonparametric segmentation of curves into various representations. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 17(12):1140–1153, 1995.

[110] G. Roth and M. D. Levine. Geometric primitive extraction using a genetic algorithm. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 16(9):901–905, September 1994.

[111] C. A. Rothwell. *Object Recognition through Invariant Indexing*. PhD Thesis, Department of Engineering Science, University of Oxford, 1994.

[112] C. A. Rothwell, A. Zisserman, D. A. Forsyth, and J. L. Mundy. Planar object recognition using projective shape representation. *International Journal of Computer Vision*, 16(1):57–99, 1995.

[113] R. Safaee-Rad, I. Tchoukanov, B. Benhabib, and K. C. Smith. Accurate parameter estimation of quadratic curves from grey-level images. *CVGIP: Image Understanding*, 54(2):259–274, September 1991.

[114] P. Saint-Marc, J.-S. Chen, and G. Medioni. Adaptive smoothing: A general tool for early vision. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 13(6):514–528, 1991.

[115] P. D. Sampson. Fitting conic sections to "very scattered" data: An iterative refinement of the Bookstein algorithm. *Computer Graphics and Image Processing*, 18:97–108, 1982.

[116] J. Sato and R. Cipolla. Affine integral invariants for extracting symmetry axes. In *Proceedings, British Machine Vision Conference*, pages 63–72, 1996.

[117] H. Schwetlick and T. Schutze. Least-squares approximation by splines with free knots. *BIT*, 35(3):361–384, 1995.

[118] T. W. Sederberg, D. C. Anderson, and R. N. Goldman. Implicit representation of parametric curves and surfaces. *Computer Vision, Graphics, and Image Processing*, 28:72–84, 1984.

[119] S. S. Sinha and B. G. Schunck. Discontinuity preserving surface reconstruction. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 229–234, 1989.

[120] F. Solina and R. Bajcsy. Recovery of parametric models from range images - the case for superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 12(2):131–147, 1990.

[121] L. Stark and K. Bowyer. Acheiving generalized object recognition through reasoning about association of function to structure. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 13(10):1097–1104, 1991.

[122] F. J. Stein. *Structural Indexing for Object Recognition*. PhD thesis, School of Engineering, Univ. of Southern California, 1992.

[123] G. Taubin. Nonplanar curve and surface estimation in 3-space. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 644–645, 1988.

[124] G. Taubin. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 13(11):1115–1138, November 1991.

[125] G. Taubin. Distance approximations for rasterizing implicit curves. *ACM Transactions on Graphics*, 13(1):3–42, 1994.

[126] G. Taubin, F. Cukierman, S. Sullivan, J. Ponce, and D. Kriegman. Parametrized families of polynomials for bounded algebraic curve and surface fitting. *PAMI*, 16(3):287–303, March 1994.

[127] D. Terzopoulos. Regularisation of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Analysis*, 8(4):413–424, July 1986.

[128] P. Torr. Robust vision. In *Proceedings, British Machine Vision Conference*, 1993.

[129] M. Umasuthan and A. M. Wallace. A comparitive analysis of algorithms for fitting planar curves and surfaces defined by implicit polynomials. In R. B. Fisher, editor, *Design and Application of Curves and Surfaces*. Clarendon Press, Oxford, 1994.

[130] Richard von Mises. *Mathematical Theory of Probability and Statistics*. Academic Press, 1964.

[131] P. Whaite and F. Ferrie. Active exploration: Knowing where we're wrong. In *Proceedings, International Conference on Computer Vision*, 1993.

[132] J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Claredon Press, Oxford, England, 1965.

[133] H. K. Yuen, J. Illingworth, and J. Kittler. Detecting partially occluded ellipses using the Hough transform. *Image and Vision Computing*, 7(1):31–37, 1989.

[134] Z. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. Technical Report 2676, INRIA Sophia Antipolis, 1995.

[135] S. C. Zhu and A. L. Yuille. Region competition: Unifying snakes, region growing, and Bayes/MDL for multi-band image segmentation. In *Proceedings, International Conference on Computer Vision*, 1995.

[136] A. Zisserman, D. Forsyth, J. Mundy, C. Rothwell, J. Liu, and N. Pillow. 3D object recognition using invariance. *Artificial Intelligence*, 78(1–2):239–288, 1995.