Robust Statistics for Shape Fitting and Pattern Matching

Andrew Stein Michael Werman

Department of Computer Science,
Hebrew University,
91904 Jerusalem, Israel

Abstract

In recent years the computer vision community has been "robustifying" vision algorithms, often without a proper understanding of the formal theories behind the statistical concepts of robustness. This paper's starting point are the concepts of the breakdown point and equivariance properties of an estimator. The desired equivariance properties for shape fitting are defined, and high breakdown point methods with these properties are found. This paper includes a survey of other "robust fitting" papers, and applications of robust fitters to Minimum Length Encoding of waveforms and searching images for shapes are shown.

Introduction

In a typical computer vision task, a machine is given a set of pictures from which it has to make conclusions about the real world. All these tasks are basically huge information reduction problems. A robot with a video camera may receive millions of bytes of information each second, from which he has to reach conclusions which may be expressed in only a few bytes. *Statistics* is the formal field concerned with the reduction of lots of information (the sample) to a summarizing conclusion (the estimate). Therefore, many computer vision algorithms are analogous to statistical estimation processes. The need for statistical estimation in vision algorithms is also necessary owing to the errors intrinsic to visual signals. But these errors are not the only type of errors with which vision algorithms have to cope. Let us suppose we wish to create an program which fits shapes to a depth map. The input to such an program may be supplied by an existing "depth-from-stereo" program. The existing program estimates the 3D location of a point by applying triangulation to the corresponding points in two 2D projections. The error distribution of the 3D location of a point may be modeled if we know the nature of the errors in the 2D projections and if we have correctly matched the corresponding points of the projections. If, however, this "matching problem" has not been solved correctly by the depth-from-stereo program, then the 3D location passed to our program will be "wild", the error in the 3D location not being modeled by a known distribution function. In this case we may consider the 3D points passed to our program to consist of a sample of points with a known error distribution (points from correctly matched pairs) *contaminated* by a number of points with gross errors in their location (points from incorrectly matched pairs). A typical shape fitting algorithm chooses the shape which minimizes the sum of squares of the distances of the 3D points to the shape. This is correct if these distances (i.e., errors) are normally distributed with zero mean, but because of the contamination by gross errors, minimizing the sum of squares of the distances is no longer reasonable, and a more *robust* method is needed.

Robust Statistics

Robust statistics deals with the fact that many of the common assumptions made in statistics (such as normality, independence, accurate sampling process) are not always correct in real situations. Classical statistical procedures are generally optimal under exact models of the error distributions, but are unstable under small deviations from the models. During the last three decades, formal theories, known as robust statistical theories, have been created to deal with these deviations from the *underlying models* of the error distribution ([11] [6] [9] [10] [28]).

The *breakdown point* of an estimator may be roughly defined as the smallest percentage of gross errors which may cause the estimator to take on arbitrarily large values [4]. For example, the mean of a one dimensional sample has a breakdown point $\varepsilon^* = 0$, because moving one member of the sample to ∞, will move the mean to ∞. On the other hand, the sample median has a breakdown point of $\varepsilon^* = 50\%$, because at least half the sample must be moved in order to achieve complete control over the value of the median. In the cases discussed here the maximal breakdown point is 50%. Beyond 50% we cannot differentiate between the "good" sample values and the gross errors.

The concept of the breakdown point of an estimator is easy to explain, and in vision problems it is often necessary to have a high breakdown point estimator because of the number of gross errors which occur. This paper concentrates on defining high breakdown point estimators, and in fact "robust" here generally means "high breakdown point". However, gross errors are not the only problems dealt with by robust statistics. Others are errors owing to rounding, non-independence, incorrect models, etc. A richer quantification of robustness is the *influence function*, and the derived concepts of *gross error sensitivity*, and *B-robustness* which we have discussed elsewhere [32] and which are fully defined in the above textbooks.

Among the various properties of statistical estimators, we will pay particular attention to the *equivariance* properties of the estimators we develop. To illustrate, if we are given n sample points $x_1, \ldots, x_n$, *1D location* estimates estimate the "center" of the sample. The mean and median are examples of estimates of 1D location. Any 1D location estimate T should at the minimum be *location equivariant* meaning that $T(x_1 + b, \ldots, x_n + b) = T(x_1, \ldots, x_n) + b$ for any $b \in \mathcal{R}$. This location equivariance property is implicit in the notion of a location estimate. Another equivariance properties which 1D location estimates may satisfy is *scale equivariance* — $T(cx_1, \ldots, cx_n) = cT(x_1, \ldots, x_n)$ for any constant c. This implies that the estimate is independent of the choice of measurement unit for the sample. The mean and median have both the above equivariance properties.

Linear Regression

The classical linear model is given by $y = \boldsymbol{x\beta}$, where $\boldsymbol{x} \in \mathcal{R}^p$ is a row vector of explanatory variables, $\boldsymbol{\beta} \in \mathcal{R}^p$ is a column vector of parameters and y is the response variable. This model is more powerful than may seem at first, because we allow some of the coordinates of $\boldsymbol{x}$ to be functions of others. For example if $\boldsymbol{x} = (x^{p-1}, \ldots, x, 1)$ then we have modeled the $p-1$ degree polynomial $y = \sum \beta_{p-i} x^i$. In particular if $\boldsymbol{x} \equiv (1)$, then the model is $y = \beta$ and we are dealing with 1D location, and if $\boldsymbol{x} = (x, 1)$, then the model is $y = \beta x + \gamma$ and we are dealing with *simple* linear regression. In addition, some nonlinear models may be transformed into linear models. For example, changing the scale of measurement of the response variable in the model $y = \beta_2 \exp \beta_1 x$ to a logarithmic scale leads to the simple linear model $\ln y = \beta_1 x + \ln \beta_2$.

Given the n points $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n) \in \mathcal{R}^{p+1}$, the classical *least squares* estimator finds the value $\hat{\boldsymbol{\beta}}$ minimizing $\sum r_i^2$, where the ith residual r_i is defined as $r_i = r_i(\boldsymbol{\beta}) = y_i - \boldsymbol{x}_i \boldsymbol{\beta}$. By setting the derivative of $\sum r_i^2$ to zero, least squares leads to an explicit formula for $\hat{\boldsymbol{\beta}}$. If we consider p to be a constant, calculating $\hat{\boldsymbol{\beta}}$ using this formula needs $O(n)$ operations. If the actual errors occur only in the response variable and are normally distributed with mean zero and unknown standard deviation σ, then the least squares estimator is the "best" estimator according to many statistical criteria. However, if the error process is not Gaussian, or the sample is contaminated by outliers, or if errors occur in the explanatory variables as well as the response variable, the least squares estimator can give vary aberrant estimates. Indeed, a large error in one of the y_i can cause the estimates to take infinitely large values. In other words the least squares estimator has a breakdown point of $\varepsilon^* = 0\%$.

The *least absolute values* or L_1 estimator takes the value of $\hat{\boldsymbol{\beta}}$ minimizing $\sum |r_i|$. This is a generalization of the median of a one dimensional sample. Least absolute values regression is more robust than least squares in the sense that it can protect against outlying response variables, however, it cannot cope with grossly incorrect values of $\boldsymbol{x}_i$. In fact, except for the 1D location case (no explanatory variables), least absolute values regression also has a breakdown point of $\varepsilon^* = 0\%$.

M-estimators minimize $\sum \rho(r_i)$, where ρ is a symmetric function with a unique minimum at zero. These include least squares ($\rho(t) = \frac{1}{2}t^2$) and least absolute values ($\rho(t) = |t|$). Differentiating $\sum \rho(r_i)$ with respect to $\boldsymbol{\beta}$ and setting to zero yields the p equations $\sum \psi(r_i)\boldsymbol{x}_i = 0$, where ψ is the derivative of ρ. However, unlike least squares and least absolute values regression, for other choices of ρ and ψ the solution these equations is not equivariant with respect to magnification of the error scale, therefore, one has to standardize the residuals by some estimate $\hat{\sigma}$ of the scale parameter, yielding the equations

$$\sum_{i=1}^{n} \psi(r_i/\hat{\sigma})\boldsymbol{x}_i = 0. \tag{1}$$

If ψ is bounded (for example in the case of least absolute values), the M-estimator can guard against outlying values of y_i. This is all that is necessary if we are dealing with 1D location. In fact if ψ is bounded and $\hat{\sigma}$ is found with a 50% breakdown point estimator, then the 1D M-estimate of location has a breakdown point of 50%. However, in the general case the M-estimator cannot guard against outlying $\boldsymbol{x}_i$.

In addition to least squares and least absolute values, M-estimators include

- *Huber's minimax* — $\rho(t) = \begin{cases} \frac{1}{2}t^2 \\ c|t| - \frac{1}{2}c^2 \end{cases} \quad \psi(t) = \begin{cases} t & |t| \leq c \\ c \operatorname{sgn} t & |t| > c, \end{cases}$
  which treats small values of r_i like least squares and values larger than c like least absolute squares. With this we achieve the robustness of least absolute values without sacrificing all the statistical advantages of least squares at the normal distribution.
- *Tukey's biweight* — $\rho(t) = \begin{cases} \frac{c^2}{6}\left(1 - \left(1 - (\frac{t}{c})^2\right)^3\right) \\ \frac{c^2}{6} \end{cases} \quad \psi(t) = \begin{cases} t\left(1 - (\frac{t}{c})^2\right)^2 & |t| \leq c \\ 0 & |t| > c, \end{cases}$
  which is a member of the class known as *redescending estimators* because ψ comes back to zero when the absolute value of its argument is greater than a specified positive number.

One of the most commonly used robust estimators of scale (see [12] for a survey) is the *median absolute deviation* from the median defined as $\hat{\sigma} = 1.482 \operatorname{med}_i |r_i - \operatorname{med}_j r_j|$, 1.482 being the correction for the median absolute deviation for samples from the standard normal distribution. Another approach is to use a M-estimator of scale, which yield an additional equation $\sum \chi(r_i/\hat{\sigma}) = 0$, where χ is an even function.

M-estimators are found using iterative minimization and equation solving algorithms [18]. With any iterative value, a good starting value is important, especially when ρ is not convex and ψ redescends. Below we describe another use of the high breakdown estimators as starting values for M-estimators.

The *Theil* estimator [34] is defined as follows. Each subset $I = \{i_1, \ldots, i_p\}$ of $\{1, \ldots, n\}$ containing p indices defines a parameter vector $\boldsymbol{\beta}(I) = \boldsymbol{\beta}(i_1, \ldots, i_p)$. If $\beta_j(I)$ is the jth coordinate of $\boldsymbol{\beta}(I)$, the jth coordinate of $\hat{\boldsymbol{\beta}}$ is

$$\hat{\beta}_j = \operatorname*{med}_I \beta_j(I), \tag{2}$$

where the median is over all the $I = \{i_1, \ldots, i_p\}$ for which $\boldsymbol{\beta}(I)$ is defined. The computational complexity of finding $\hat{\boldsymbol{\beta}}$ is large, namely $O(n^p)$, because all the $\binom{n}{p}$ vector parameters $\boldsymbol{\beta}(I)$ need to be found. In addition the breakdown point of this estimator is $\varepsilon^* = 1 - 1/\sqrt[p]{2}$, which for moderate p is very low. However, for simple linear regression an $O(n \log n)$ algorithm exists for finding $\hat{\boldsymbol{\beta}}$ [3], which together with the 29% breakdown point makes this estimator practicable.

The jth coordinate of Siegel's *repeated median* estimator [30], is defined by

$$\hat{\beta}_j = \operatorname*{med}_{i_1} \operatorname*{med}_{i_2} \cdots \operatorname*{med}_{i_p} \beta_j(i_1, \ldots, i_p), \tag{3}$$

where the innermost median is over all the $\beta_j(i_1, \ldots, i_p)$ which are defined. This estimator does achieve a 50% breakdown point, however, the complexity is still $O(n^p)$. For simple linear regression we have developed an $O(n(\log n)^2)$ algorithm for calculating this estimator [32].

Another estimator having a 50% breakdown point is the *least median of squares* estimator [27], defined as the value of $\boldsymbol{\beta}$ minimizing $\operatorname{med}_i r_i^2$. If we are dealing with 1D location, least median of squares estimate is the center of the shortest interval containing half the given points, which can be found in $O(n \log n)$ operations.

For simple linear regression this estimator is the narrowest strip covering half the given points, where the thickness of the strip is measured vertically. There is an $O(n^2)$ algorithm for simple regression [31], but for more complicated cases no exact algorithm is known. In practice [28] an approximation algorithm, which has a complexity of $O(n)$, is used.

The major drawback of the least median of squares and repeated median estimators is their low efficiency. The convergence rate of the least median of squares estimator is $n^{-1/3}$ as opposed to the convergence rate of $n^{-1/2}$ of the M-estimators The efficiency of the repeated median estimator appears to be unknown. In practice these estimators are often used as starting values for calculating M-estimators with non-convex ρ. In this case it is necessary to use the least median of squares estimator, and not the repeated median estimator, so that the calculation of the starting values is not of orders of magnitude more complex than finding the M-estimator itself.

The more important equivariance properties associated with a linear regression estimator T are

- *Regression equivariance* — $T\left((\boldsymbol{x}_1, y_1 + \boldsymbol{x}_1\boldsymbol{\gamma}), \ldots, (\boldsymbol{x}_n, y_n + \boldsymbol{x}_1\boldsymbol{\gamma})\right) = T\left((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\right) + \boldsymbol{\gamma}$, for any p-dimensional column vector $\boldsymbol{\gamma}$. Regression equivariance for a regression estimator is just as crucial as location equivariance for a location estimator, and is implicit in the notion of a regression estimator.

- *Scale equivariance* — $T\left((\boldsymbol{x}_1, cy_1), \ldots, (\boldsymbol{x}_n, cy_n)\right) = cT\left((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\right)$, for any constant c. This implies that the fit is not dependent on the measurement unit of the response variable.

- *Affine equivariance* — $T\left((\boldsymbol{x}_1\boldsymbol{A}, y_1), \ldots, (\boldsymbol{x}_n\boldsymbol{A}, y_n)\right) = \boldsymbol{A}^{-1}T\left((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\right)$, for any $p \times p$ non-singular matrix $\boldsymbol{A}$. Affine equivariance means that a linear transformation of the $\boldsymbol{x}_i$ should transform the estimate accordingly.

All the estimators defined in this section are regression and scale equivariant. All those which minimize a function of the r_i (M-estimators and least median of squares) are also affine equivariant. On the other hand, because of the coordinate wise manner in which the Theil and repeated median estimators calculate the parameter vector $\boldsymbol{\beta}$, neither are affine equivariant.

Line Fitting

This section discusses line fitting in the *image plane*, that is, given a set of n points, $(x_1, y_1), \ldots, (x_n, y_n)$ in the image plane, we wish to find a line passing "closest" to the points. In the image plane vertical lines may appear, therefore, it is incorrect to perceive y as a response variable as in the previous section. We should demand that the method of line fitting should treat both the x and y axes equally.

One method used in computer vision which does treat x and y equally is the *median of intercepts* method [13]. In this method the line is parameterized as $x/u + y/b = 1$, u and b being, respectively, the x-axis and y-axis intercepts of the line. If we denote by u_{ij} and b_{ij} the intercepts of the line passing through

(x_i, y_i) and (x_j, y_j) then the line fitted by the median of intercepts method has the parameters $\hat{u} = \text{med}_{i<j}\, u_{ij}$ and $\hat{b} = \text{med}_{i<j}\, b_{ij}$. However, this method may fail when the line is almost horizontal or almost vertical. If, for example, the line is almost horizontal, the value of u_{ij} will be "close" to $\pm\infty$ if both (x_i, y_i) and (x_j, y_j) are "good" points. The situation may arise where nearly half of the u_{ij} are near $+\infty$ and nearly half of the u_{ij} are near $-\infty$. The median of the u_{ij} will then be determined by the few which arise from the erroneous points, but have values in the middle of the set. A similar problem occurs with almost vertical lines as can be seen in the figure at the end of this section.

Even though the median of intercepts method is rather robust, we see that it is incapable of coping with an almost vertical or horizontal line. It is not enough to treat the x and y axes equally for images. We should demand the stronger properties that whenever the points are rotated and/or translated the fitted line will be rotated and/or translated by the same amount. We will call these properties *rotation* and *translation equivariance* respectively. For any method in which y is seen as a function of x, i.e., the line is modeled as $y = ax + b$, rotation equivariance will never hold. Similarly, rotation equivariance does not hold for the parameterization $x/u + y/b = 1$ which is unable to cope with horizontal or vertical lines. In addition to the properties of rotation and translation equivariance, we will also prefer our estimator to be *scale equivariant*, meaning that if the measurement unit of the x- and y-axes changes *equally* (i.e., $(x, y) \longrightarrow (cx, cy)$, $c \neq 0$) the fitted line will change appropriately (similarity transformations). Our estimators will also be *reflection equivariant*, although we will not demand this property.

One method of achieving these properties is not to use the residuals r_i to define the distance of the point (x_i, y_i) from a line. Rather define d_i as the (signed) length of the perpendicular of the point to the line, which is invariant to rotations and translations of the image plane. We may then minimize $\sum d_i^2$, $\sum |d_i|$, $\sum \rho(d_i)$ or $\text{med}\, d_i^2$. Because d_i is invariant to translations and rotations, these minimizations are rotation and translation equivariant. It is easily shown [32] that minimizing $\sum d_i^2$, $\sum |d_i|$ or $\text{med}\, d_i^2$ is scale equivariant. Minimizing $\sum \rho(d_i)$ can also be made scale equivariant by standardizing the perpendicular residuals d_i by a robust estimate of scale.

If $\sum d_i^2$ is to be minimized we are using the method called *eigenvector* line fitting [5]. This line can be found in $O(n)$ time but the fit is not robust (has a 0% breakdown point). If $\sum |d_i|$ or $\sum \rho(d_i)$, with a proper choice of ρ, is minimized, the estimator will be more robust than the eigenvector line fit, but we will still suffer from a 0% breakdown point. However, minimizing $\text{med}_i\, d_i^2$ is a robust method having a breakdown point of $\varepsilon^* = 50\%$ as well as the desired property of equivariance under translations and rotations of the image points.

Minimizing $\sum \rho(d_i)$ is equivalent to finding the maximum likelihood estimator for the line if the distances d_i are independently distributed with a density $f(d) = k_1 \exp(-k_2 \rho(d))$, where k_1 and k_2 are positive numbers not dependent on d. Weiss [35] in fact reached such an estimator by modeling the image points as coming from two sources: the "genuine" data which are distributed normally, and the "noise" which is distributed uniformly in the image area. Taking the maximum likelihood estimator for this mixed normal and uniform distribution, he arrived at a set of equations similar to those which minimize $\sum \rho(d_i)$. The ρ function reached is not convex and he had difficulties arising from local extrema.

From the discussion above we see that the line fitting method's correctness is directly dependent on the choice of parameters used to describe the line. The $y = ax + b$ parameterization cannot describe vertical lines. The $x/u + y/b = 1$ parameterization cannot describe horizontal, vertical or lines through the origin. We prefer the *normal equation* of the line

$$l = x\cos\varphi + y\sin\varphi. \tag{4}$$

We call φ the *normal direction* and l the *normal length*. We also choose $\cos\varphi \geq 0$ because under this choice the (signed) perpendicular distance of a point (x_i, y_i) from the line parameterized by (l, φ) is $d_i = l - x_i\cos\varphi - y_i\sin\varphi$, which simplifies the minimization of $\sum \rho(d_i)$ or $\operatorname{med} d_i^2$.

The important thing to understand when using the normal parameters is that the normal direction is not a linear parameter, but is in fact a *directional* parameter. We understand values of φ slightly less than $\pi/2$ to be close to values slightly greater than $-\pi/2$. In other words values of $\theta = 2\varphi$ should be looked upon as values on the unit circle and not as values on the real line. This is the crucial reason for the failure of the median of intercepts method, because the linear median operator treats intercept values near $+\infty$ as very different from values near $-\infty$, even though lines with these intercepts may be very close.

If we wish to adapt the Theil or repeated median estimators to computer vision line fitting, it is necessary to define a concept similar to the median, which is valid for directional values. The *circular median* of $\theta_1, \ldots, \theta_n$ is defined [19] as any point ξ on the unit circle satisfying the following conditions: (i) half of the given points are on either side of the diameter from ξ to $\xi + \pi$ and (ii) the majority of the points are nearer to ξ than $\xi + \pi$, is called the circular median of $\theta_1, \ldots, \theta_n$. The circular median of n points can be found in time $O(n)$ using an algorithm described elsewhere [32].

Let us denote the normal parameters of the line through (x_i, y_i) and (x_j, y_j) by (l_{ij}, φ_{ij}). We define the normal direction of the *circular Theil* estimator as

$$\hat{\varphi} = \frac{1}{2} \operatorname*{circmed}_{i<j} 2\varphi_{ij}. \tag{5}$$

We have shown [32] an $O(n(\log n)^2)$ algorithm for finding the normal direction of the circular Theil estimator. The normal direction of the *repeated circular median* estimator is similarly defined as

$$\hat{\varphi} = \frac{1}{2} \operatorname*{circmed}_{i} \operatorname*{circmed}_{j \neq i} 2\varphi_{ij}. \tag{6}$$

Once $\hat{\varphi}$ has been found (either circular Theil or repeated circular median), the normal length may be estimated hierarchically as

$$\hat{l} = \operatorname*{med}_{i}(x_i\cos\hat{\varphi} + y_i\sin\hat{\varphi}) \tag{7}$$

using the linear median operator. We have proven the following lemma [32].

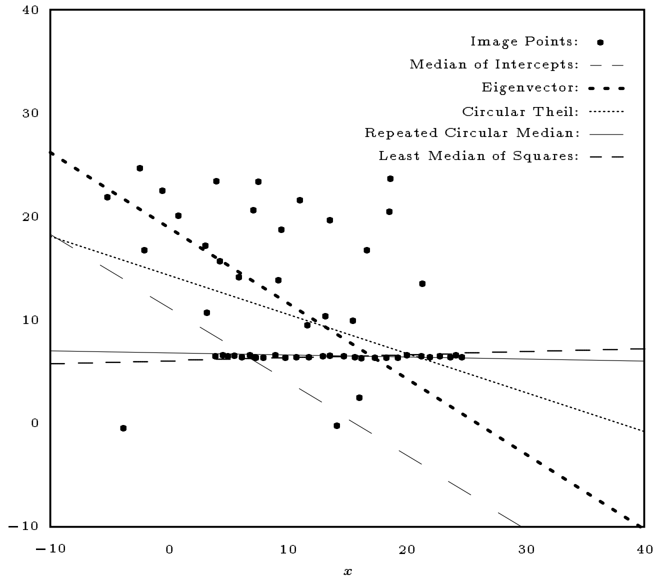

Figure 1: The repeated circular median compared to the eigenvector, circular Theil, and (regular) least median of squares fits.

Lemma 1 *The hierarchical circular Theil and repeated circular median estimates are rotation, translation and scale equivariant.*

The only question remaining is: “How robust are these estimators?” The linear Theil and repeated median estimators inherited their 29.3% and 50% breakdown points respectively, from the 50% breakdown point of the linear median. We would like to find the breakdown point of the circular median. Unfortunately, the definition of the breakdown point does not apply to directional data. The circular mean and median can be at most 180 (and not ∞) degrees wrong! Attempts to define a circular breakdown point as the percentage of gross errors needed to cause the estimator to take on values wrong by a certain amount, do not conform with the statisticians' experience that the circular mean is not robust, while the circular median is. This issue has been discussed more fully in our previous paper [32]. We have concluded that even though we do not have a notion of the breakdown point for directional data, the circular Theil and repeated circular median estimators are robust.

Figure 1 shows some of the line fitting methods discussed in this chapter, as compared to the repeated circular median estimate. In this figure, there are 54 image points consisting of 28 inliers on an almost horizontal line and 26 outliers. It can be seen that the repeated circular median estimate gives the “correct” line, as does the regular least median of squares estimate. This confirms our claim that the repeated circular median is robust.

Fitting 2D Conic Sections

This section discusses the fitting of 2D conic sections to n given points $(x_1, y_1), \ldots, (x_n, y_n)$ in the plane. We would like our fitting method to be robust as well as equivariant with respect to rotations, translations and equal scaling of the plane.

The general parameterization of a 2D conic section is by a six-dimensional parameter vector $\boldsymbol{\beta} = (A, B, C, D, E, F)^t \in \mathcal{R}^6$ as $Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$, where not all three parameters A, B and C are zero. However, $\boldsymbol{\beta}$ and $k\boldsymbol{\beta}$ describe the same conic section for any $k \neq 0$, therefore this parameterization will have to be normalized in order to get rid of this redundancy.

Let us denote by $Q(x,y) = Q(x,y;\boldsymbol{\beta})$ the quadratic polynomial $Q(x,y) = (x^2, xy, y^2, x, y, 1)\boldsymbol{\beta} = Ax^2 + Bxy + Cy^2 + Dx + Ey + F$. The conic section is then $Q(x,y) = 0$. Let us denote by $d_i = d_i(\boldsymbol{\beta})$ the (signed) Euclidean distance of the point (x_i, y_i) from this conic section. The obvious generalization of M- and least median of squares estimators is to find the value of $\boldsymbol{\beta}$ minimizing $\sum \rho(d_i)$ and $\operatorname{med} d_i^2$, respectively. Obviously these estimators are rotation and translation equivariant, because rotation and translation of the plane does not change the value of d_i. In addition, equal scaling of the plane by $c \neq 0$ will change d_i to cd_i (as in the case of the line), meaning that the least median of squares estimator is scale equivariant, and that the M-estimator can be made scale equivariant with an appropriate standardization by an estimate $\hat{\sigma}$ of the error scale. The major problem is how to calculate $d_i(\boldsymbol{\beta})$ easily from $\boldsymbol{\beta}$. If (x_0, y_0) is the closest point to (x_i, y_i) on the conic section, then (x_0, y_0) can be found by simultaneously solving two quadratic equations in two unknowns, which may have more than one solution. In order to avoid the problems encountered when trying to solve these equations, which have to be solved for every i for every $\boldsymbol{\beta}$ encountered, we will try using an easily computed approximation of d_i. However, when using an approximation, we will have to make sure that our fitting procedure is still rotation, translation and scale equivariant.

Bookstein [2] takes $\delta_i(\boldsymbol{\beta}) = Q(x_i, y_i; \boldsymbol{\beta})$ as an approximation of $d_i(\boldsymbol{\beta})$, and minimizes $\sum \delta_i^2$ using the normalizing constraint $A^2 + B^2/2 + C^2 = 2$. This leads to an easily computable problem in eigenanalysis. In [32] we prove that the Bookstein fit is rotation, translation and scale equivariant. However, because $\sum \delta_i^2$ is minimized, the Bookstein method is not robust and therefore we prefer to minimize $\sum \rho(\delta_i)$ or $\operatorname{med} \delta_i^2$. Both of these are translation and rotation equivariant and minimizing $\operatorname{med} \delta_i^2$ is also scale equivariant. In order to make minimizing $\sum \rho(\delta_i)$ scale equivariant, we have to standardize the δ_i by *the square of* a robust estimate of scale.

Another problem associated with the Bookstein distances is that they are not very good approximations as can be seen is Figure 2 and Table 1. The Bookstein distances were chosen in order to make the fit equivariant and simple to calculate. If we decide to minimize $\sum \rho(\delta_i)$ or $\operatorname{med} \delta_i^2$, the calculation looses its simplicity and we might as well use a better approximation.

To improve the approximation to the actual distance, Sampson [29] suggests taking $\Delta_i = \Delta_i(\boldsymbol{\beta}) = Q(x_i, y_i; \boldsymbol{\beta}) / \|\nabla Q(x_i, y_i; \boldsymbol{\beta})\|$ as the approximation and minimizing $\sum \Delta_i^2$. Sampson minimizes $\sum \Delta_i^2$ by

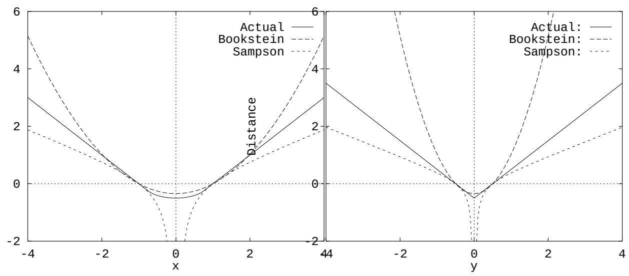

Figure 2: Distances of the point $(x, 0)$ (left) and $(0, y)$ (right) from the ellipse $x^2 + 4y^2 = 1$.

Distance of	$(x, 0)$	$(0, y)$
Actual	$\begin{cases} \lvert x\rvert - 1 & \lvert x\rvert \geq \frac{3}{4} \\ -\sqrt{\frac{1}{4} - \frac{x^2}{3}} & \lvert x\rvert < \frac{3}{4} \end{cases}$	$\lvert y\rvert - \frac{1}{2}$
Bookstein	$\sqrt{\frac{2}{17}}(x^2 - 1)$	$\sqrt{\frac{2}{17}}(4y^2 - 1)$
Sampson	$\frac{x^2-1}{2\lvert x\rvert}$	$\frac{4y^2-1}{8\lvert y\rvert}$
Nalwa-Pauchon	$\lvert x\rvert - 1$	$\lvert y\rvert - \frac{1}{2}$

Table 1: Distance functions of Figure 2.

using an iterative procedure starting with the Bookstein fit as an initial guess.

It is easy to show that the Sampson fit is rotation, translation and scale equivariant, but the fit is not robust. If we replace $\sum \Delta_i^2$ with $\sum \rho(\Delta_i)$, a proper choice of ρ will ensure that the fit will be robust as well as rotation, translation and scale (after standardizing by an estimate of scale) equivariant. Minimizing $\text{med}_i\, \Delta_i^2$ is of course even more robust as well as having all the mentioned equivariance properties. The main problem with the Sampson distance Δ_i is that it approaches infinity as (x_i, y_i) approaches the center of the conic. This will cause any procedure based on Sampson distances to penalize points nearer the center of the conic section much more heavily than points on the "outside" of the conic section. This may in turn will tend to warp conic sections estimated by Sampson distances depending on the type of minimization used.

The problems which arise when fitting conic sections using "distance based" estimators, such as the M- and least median of squares estimates, encourage one to look into "parameter based" estimators, even though these need more time to calculate.

Every five non-correlated points $(x_{i_1}, y_{i_1}), \ldots, (x_{i_5}, y_{i_5})$ generate a parameter vector $\boldsymbol{\beta}(i_1, \ldots, i_5)$. Here we speak of a parameter vector after normalization, therefore $\boldsymbol{\beta}(i_1, \ldots, i_5)$ is a five dimensional vector, whose jth coordinate, $1 \leq j \leq 5$, is $\beta_j(i_1, \ldots, i_5)$. The Theil estimator is generalized to this situation by taking $\hat{\beta}_j = \mathrm{med}_{i_1<\cdots<i_5} \beta_j(i_1, \ldots, i_5)$ as the jth coordinate of the estimated vector $\hat{\boldsymbol{\beta}}$. The repeated median estimator will choose $\hat{\beta}_j = \mathrm{med}_{i_1} \mathrm{med}_{i_2} \cdots \mathrm{med}_{i_5} \beta_j(i_1, \ldots, i_5)$ as the jth coordinate of $\hat{\boldsymbol{\beta}}$. The time needed to calculate these estimates is $O(n^5)$, which may be too long for many applications.

The central issue here is the normalization of the parameter vector $(A, B, C, D, E, F)^t$. This is important because if we are not careful in our choice of normalization, the coordinate wise method in which the Theil and repeated median estimators are calculated will cause the estimators not to have the desired equivariance properties. In [32] we show that we cannot make these estimators rotation equivariant using the normalization $A^2+B^2/2+C^2 = 2$. The problem is to find more "natural" parameters to describe a conic section: parameters which change independently of each other under rotation, translation and scaling of the plane. We suggest using the geometric parameters of the center $(\zeta, \eta) \in \mathcal{R}^2$, axes $(a, b) \in \mathcal{R}^2_+$ and angle φ of the major axis, that is the parameterization

$$\frac{((x-\zeta)\cos\varphi + (y-\eta)\sin\varphi)^2}{a^2} \;\square\; \frac{(-(x-\zeta)\sin\varphi + (y-\eta)\cos\varphi)^2}{b^2} = 1, \tag{8}$$

where $\square$ is $+$ for an ellipse or $-$ for a hyperbola, and for ellipses $a > b$. We note that the center of the conic is in fact a *two dimensional location* value, therefore ζ and η should be estimated together using the 2D spatial median. In addition φ being an axial value should be estimated using the circular median. The Theil estimator should then become

$$\begin{array}{rcl} (\hat{\zeta}, \hat{\eta}) & = & \mathrm{spatmed}_{i_1<\cdots<i_5}(\zeta, \eta)(i_1, \ldots, i_5) \\ \hat{a} & = & \mathrm{med}_{i_1<\cdots<i_5}\, a(i_1, \ldots, i_5) \\ \hat{b} & = & \mathrm{med}_{i_1<\cdots<i_5}\, b(i_1, \ldots, i_5) \\ \hat{\varphi} & = & \frac{1}{2}\,\mathrm{circmed}_{i_1<\cdots<i_5}\, 2\varphi(i_1, \ldots, i_5) \end{array} \tag{9}$$

and the repeated median estimator should become

$$\begin{array}{rcl} (\hat{\zeta}, \hat{\eta}) & = & \mathrm{spatmed}_{i_1}\, \mathrm{spatmed}_{i_2} \cdots \mathrm{spatmed}_{i_5}(\zeta, \eta)(i_1, \ldots, i_5) \\ \hat{a} & = & \mathrm{med}_{i_1}\, \mathrm{med}_{i_2} \cdots \mathrm{med}_{i_5}\, a(i_1, \ldots, i_5) \\ \hat{b} & = & \mathrm{med}_{i_1}\, \mathrm{med}_{i_2} \cdots \mathrm{med}_{i_5}\, b(i_1, \ldots, i_5) \\ \hat{\varphi} & = & \frac{1}{2}\,\mathrm{circmed}_{i_1}\, \mathrm{circmed}_{i_2} \cdots \mathrm{circmed}_{i_5}\, 2\varphi(i_1, \ldots, i_5). \end{array} \tag{10}$$

In order to save time, one may estimate some of the coordinates of $\boldsymbol{\beta} = (\zeta, \eta, a, b, \varphi)^t$ hierarchically from others. In [32] we prove

Lemma 2 *The Theil (9) and repeated median (10) as well as hierarchical variants of (9) and (10) are rotation, translation and scale equivariant.*

Figure 3 shows some of the methods described in this chapter as applied to circle fitting. In this figure there are 15 image points, 5 of which are gross outliers. The two least squares methods, one based on the actual distances (which are easy to calculate for a circle) and the other based on the Bookstein method, are both incorrect, while the Theil estimator, while better, is still over influenced by the outliers. The repeated median gives a much better estimate.

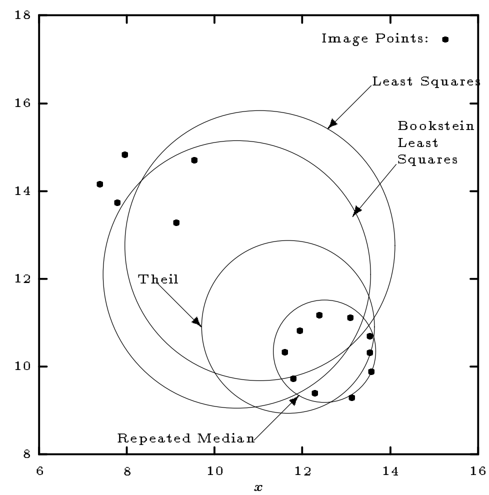

Figure 3: The repeated median compared to the actual least squares, Bookstein least squares and Theil fits.

Shape Fitting

This section generalizes the ideas developed in the previous sections to general shape fitting. The shape is modeled by an implicit equation $f(\boldsymbol{x};\boldsymbol{\beta}) = 0$, where $\boldsymbol{\beta} \in \mathcal{R}^q$ is the sought parameter vector describing the shape which "best" fits the n given points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{R}^q$. The shape estimator should be selected according to the following criteria:

- the desired equivariance properties
- the robustness and efficiency
- the complexity of the calculation involved

We divide the estimators into two groups: *distance based* estimators which minimize a function of a measure of the distances of the given points to the shape and *parameter based* estimators which calculate a function of the parameter vectors defined by each q-tuple of input points. The choice between the two groups should be based on the above criteria.

For the distance based estimators we have discussed, namely the M- and least median of squares estimators, if actual (Euclidean) distance measures are used, there is no problem with the rotation and equivariance properties and the scale equivariance problem with M-estimators can be solved with the scale standardization we have seen in previous sections. M-estimates in this context have a breakdown point of $\varepsilon^* = 0\%$, as opposed to the 50% breakdown point of the least median of squares estimator, but with a proper choice of ρ they are more efficient. The choice of ρ is base on the underlying distribution (before contamination) of the errors. The efficiency of least median of squares estimation should be improved by "one-step M-estimation" or "reweighted least squares" as described elsewhere [32]. The main problem with the distance based methods is that the actual distance $d(\boldsymbol{\beta})$ of $\boldsymbol{x}$ from $f(\boldsymbol{x};\boldsymbol{\beta}) = 0$ may not be simple to calculate. In many cases, finding the closest point $\boldsymbol{x}_0$ to $\boldsymbol{x}$ for which $f(\boldsymbol{x}_0;\boldsymbol{\beta}) = 0$ may not lead to explicit equations for $\boldsymbol{x}_0$ or the explicit equations may have more than solution. Implicit equations may need an iterative procedure to solve. All this, at best, adds to the complexity of the algorithm, which may in itself require iterative procedures to calculate. Therefore in some cases approximations to the actual distances are used. If approximations are used it is necessary to ensure that the equivariance properties continue to hold.

The generalization of the Bookstein distance used for conic sections is to take $\delta = f(\boldsymbol{x}) = f(\boldsymbol{x};\boldsymbol{\beta})$ as the approximation of the distance of $\boldsymbol{x}$ from the shape. However, in the general case there is no knowing how δ behaves as a function of the actual distance. In addition, minimization of a function of the approximate distances $f(\boldsymbol{x}_1;\boldsymbol{\beta}),\ldots,f(\boldsymbol{x}_n;\boldsymbol{\beta})$ is not necessarily equivariant. (It was the normalization $A^2+B^2/2+C^2 = 2$ which made the estimates of the previous section based on the Bookstein distance equivariant.)

On the other hand, the generalization of the Sampson distance does have a formal justification. The first order approximation to the value of $f(\boldsymbol{x}_0)$ is $f(\boldsymbol{x}_0) \approx f(\boldsymbol{x}) + \langle \boldsymbol{x}_0 - \boldsymbol{x} \ , \ \nabla f(\boldsymbol{x})\rangle$. If $\boldsymbol{x}_0$ is the closest point to $\boldsymbol{x}$ on the shape, the above approximation becomes

$$-f(\boldsymbol{x}) \approx ||\boldsymbol{x}_0 - \boldsymbol{x}|| \cdot ||\nabla f(\boldsymbol{x})|| \cos\psi \Longrightarrow ||\boldsymbol{x}_0 - \boldsymbol{x}|| \approx \frac{-f(\boldsymbol{x})}{||\nabla f(\boldsymbol{x})|| \cos\psi}, \tag{11}$$

where ψ is the angle between $\boldsymbol{x}_0 - \boldsymbol{x}$ and $\nabla f(\boldsymbol{x})$. Because $\boldsymbol{x}_0$ is the closest point on $f(\boldsymbol{x}) = 0$ to $\boldsymbol{x}$, the gradient of f at $\boldsymbol{x}_0$ is parallel to $\boldsymbol{x}_0 - \boldsymbol{x}$. If we assume that the gradient of f at $\boldsymbol{x}$ has approximately the same direction as the gradient of f at $\boldsymbol{x}_0$, then $\cos\psi \approx 1$. Using this second approximation together with the above, we reach the generalized Sampson distance, which is (with a change of sign) $\Delta = f(\boldsymbol{x})/||\nabla f(\boldsymbol{x})||$. The norm of the gradient of a function does not change under translations and rotations of $\mathcal{R}^p$. This means that the minimizations of functions of the Sampson distances of the given points to the shape parameterized by $\boldsymbol{\beta}$, is rotation and translation equivariant if the value of f (the Bookstein distance) does not change under rotation and translation of $\mathcal{R}^p$. The norm of the gradient is rescaled by $|c|$ under a scaling of $\mathcal{R}^p$ by $c \neq 0$. Depending on the value of f under scaling, minimizations of functions of the Sampson distances of the points from the shape may be standardized to be scale equivariant.

The problem of the Sampson distance approaching infinity for points near the center of the conic section, which we saw in the previous section, came from using the first order approximation to the value of $f(\boldsymbol{x}_0)$. Nalwa and Pauchon [23] used a second order approximation for fitting conics. For the general case, this comes to taking

$$||\nabla f(\boldsymbol{x})||^2 \frac{||\nabla f(\boldsymbol{x})|| \pm \sqrt{||\nabla f(\boldsymbol{x})||^2 - 2f(\boldsymbol{x})F(\boldsymbol{x})}}{\nabla f(\boldsymbol{x}) \nabla^2 f(\boldsymbol{x}) \nabla f(\boldsymbol{x})^t} \tag{12}$$

as the distance of $\boldsymbol{x}$ from the shape $f(\boldsymbol{x}) = 0$, $\nabla^2 f(\boldsymbol{x})$ being the $p \times p$ matrix whose ijth element is $\partial^2 f(\boldsymbol{x}) \,/\, \partial x_i \partial x_j$, and the sign $+$ or $-$ is chosen so that the absolute value of the distance approximation is minimal. In [32] we show the development of this approximation. Table 1 also shows the Nalwa-Pauchon distances.

Of course we may increase the accuracy of the approximations by taking higher order approximations to $f(\boldsymbol{x}_0)$. These, however, involve solving cubic, quadratic, etc. equations, and the improvement of the accuracy may be negligible compared to the inaccuracy caused by the assumption that the direction of $\nabla f(\boldsymbol{x}_0)$ is approximately equal to the direction of $\nabla f(\boldsymbol{x})$. It should be noted that even high order approximations are inaccurate for large distances, and these approximations may not even be monotone as a function of the actual distances. The shape $f(\boldsymbol{x}) = 0$ is the same as the shape $f(\boldsymbol{x}) \exp \|\boldsymbol{x}\| = 0$, which is the same as the shape $f(\boldsymbol{x}) \sin(x_1 + 2) \arctan(x_2 + 4\pi)$, but all these descriptions of the same shape yield different distance approximations.

The problems arising with the calculation of the distance necessary for distance based estimators can be overcome by using the parameter based Theil and repeated median estimators. The repeated median estimator has a 50% breakdown point but is less efficient than the Theil estimator which has a much lower breakdown point. The efficiency of these estimators should be improved by "one-step M-estimation" or "reweighted least squares". The main issue when using these estimators is the choice of the parameter $\boldsymbol{\beta}$ used to describe the shape. In order to achieve the desired translation, rotation and scale equivariance properties we recommend using geometric parameters, which change in the predictable manner under translation, rotation and scaling. Such parameters are the p-dimensional location of the shape, the angle of the shape in p-dimensional space, the size primitives of the shape, etc. These geometric parameters lead to more "natural" descriptions of the shape, as well as having the predictable variant properties under translation, rotation and scaling. As was the case in the parameterization (8) of the conic section, the coordinate wise medians of the Theil and repeated median estimator should be taken with consideration of the type of geometric parameter described by the coordinate of the parameter vector. The p-dimensional spatial median should be used on p-dimensional location parameters and the circular or spherical median should be used on circular or spherical parameters.

Survey

This section contains a survey of the robust methods used in various fitting tasks. This section has been divided into sub-sections by the topic of the papers. The division is rather arbitrary and some sections overlap.

Window Operators

One of the first papers in the computer vision literature to incorporate robust statistical methods is Besl *et al.*'s paper [1] dealing with the robustification of the window operators used for local smoothing, derivative estimation, edge detection, etc. They used M-estimators to fit constant, planar, quadratic and cubic polynomials to a rectangular window around each pixel, stopping if one of the fits yield a zero median of absolute deviation, and choosing the best fit otherwise. The constant of each best fitting polynomial replaces the corresponding pixel for smoothing, and the derivatives of each best fitting polynomial are used as estimates for the derivatives at the pixel. Among other applications, the paper by Kim *et al.* [15] includes a description of a least median of squares based robust local operator. They show cases where their operator is preferable to an M-estimate based operator and show the necessity of using reweighted least squares to improve the efficiency of the least median of squares estimate.

Segmentation and Edge Detection

Meer and Mintz [20] together with Rosenfeld [21] [22] use a least median of squares estimator to segment depth images into a piecewise polynomial surface representation. A window of a depth image may contain pixels belonging to more than one surface. A least median of squares operator in such a window may be able to discriminate between two surfaces by marking the pixels of one surface as inliers and the pixels of the other as outliers. The image is tessellated with (rather large) windows. Inside each window some of the pixels have been marked as outliers by a least median of squares fit. At each stage the tessellation is modified by trying to extend the connected inlier pixels to include outlying pixels from neighboring regions, and by fusing neighboring regions having a similar fit.

Roth and Levine [26] used least median of squares estimators to segment range images by the following two-step iterative procedure. First the image is segmented using a threshold based step- and roof-edge detector, the segments being connected regions bounded by edge pixels and not containing pixels already segmented. Next the largest connected set of pixels is sent as the input to the robust fitter, which fits an implicit planar equation to the pixels using the least median of squares fit on the Sampson distances from the plane. If the error of the fit (based on the median of absolute deviations) is less than a certain value, the inlier pixels of the set are assigned to the fitted plane and removed from the image; otherwise a different segmentation is obtained by decreasing the threshold, and the process is repeated. If for a certain region there is no success at any threshold, the whole process is repeated using an implicit quadratic equation.

In a later paper [25], Roth and Levine provide a unifying framework, showing that the robust fitting methods [1] [15] [26] and the traditional computer vision method of the Hough transform can be understood as different manifestations of a more general model. This paper also points out that the 50% breakdown point, considered to be the maximum for statistical methods, may be as large as 80% or more in real computer vision problems. This paper has an important overview on the application of statistical methods in computer vision.

Pose Estimation

Given n 3D model points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ and the corresponding 2D perspective points on the image plane $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$, the problem of *2D–3D pose estimation* is to find the rotation matrix $\boldsymbol{R}$ and the translation vector $\boldsymbol{t}$, such that $\boldsymbol{y}_i = \boldsymbol{f}(\boldsymbol{x}_i\boldsymbol{R}+\boldsymbol{t})$, $i = 1, \ldots, n$, $\boldsymbol{f}$ being the perspective projection function. Because of noise in the measurement process and because of mismatching, no single $\boldsymbol{R}$ and $\boldsymbol{t}$ can be found which satisfies the above for all i. The least squares method takes the rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$ minimizing $\sum \|\boldsymbol{y}_i - \boldsymbol{f}(\boldsymbol{x}_i\boldsymbol{R}+\boldsymbol{t})\|^2$. This may be sufficient if there are no gross errors owing to mismatching. Haralick and Joo [7] use M-estimators for this problem, that is they take the $\boldsymbol{R}$ and $\boldsymbol{t}$ minimizing $\sum \rho\left(\|\boldsymbol{y}_i - \boldsymbol{f}(\boldsymbol{x}_i\boldsymbol{R}+\boldsymbol{t})\|/\hat{\sigma}\right)$, where $\hat{\sigma}$ is the median absolute deviation estimate of error scale. The ρ functions used were Huber's minimax (with $c = 1.5$) and Tukey's biweight (with $c = 6$). Kumar and Hanson [17] compare the 2D–3D pose estimation method described above to the equivalent method based on a least median of squares estimator, defined in an earlier paper [16], and to the traditional least squares methods. They discuss the conditions for choosing between the various options. In a later paper, Haralick *et al.* [8] simplify the above method to 2D–2D (the model is flat) and 3D–3D (the image is a depth image) pose estimation and generalize the method to the harder problem of 2D perspective–2D perspective (there are two 2D projections of a 3D object) pose estimation.

Curve and Contour Fitting

Stevenson and Delp [33] use M-estimators for fitting curves to data. In general, a curve in p-dimensional space is represented by a parametric function $\boldsymbol{f} : \mathcal{R} \longrightarrow \mathcal{R}^p$. Given the n image points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{R}^p$ the problem is to find an estimate $\hat{\boldsymbol{f}}$ of $\boldsymbol{f}$ which passes through $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. This problem is ill-posed in the sense that an infinite number of curves will fit the points. To obtain a unique and stable estimate of $\boldsymbol{f}$, *a priori* information about the properties of the curves is used in order to restrict the space of possible estimates, so that the given points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ determine a unique estimate. Generally, we demand that the estimate passes "close to" (instead of through) the points and is "smooth". For simplicity we will assume $p = 2$, that the x-coordinates of the n given points $(x_1, y_1), \ldots, (x_n, y_n)$ are in the interval $[0, 1]$ and that the curve may be expressed explicitly as a function $f : [0, 1] \longrightarrow \mathcal{R}$. The smoothness of the estimate has classically been enforced by demanding a small second derivative, by taking $\hat{f}$ as the curve minimizing $M(f) = \int_0^1 f''(x)^2 dx + \lambda \sum_{i=1}^n (f(x_i) - y_i)^2$, $\lambda > 0$ being the parameter controlling the payoff between the smoothness and the closeness of the estimate to the given points. It has been shown that the value of f minimizing $M(f)$ is a cubic spline. The major robustification proposed by Stevenson and Delp is to replace the definition of M with $M(f) = \int_0^1 \rho(f''(x))dx + \lambda \sum_{i=1}^n (f(x_i) - y_i)^2$, ρ being Huber's minimax function. This robustification allows small deviations from the smoothness constraint, that is discontinuities in the curve and the first derivative of the curve.

Applications

This section describes two of the applications we have developed for robust statistical methods in computer vision.

Minimum Length Encoding of Waveforms

Given a set of n data points $(x_1, y_1), \ldots, (x_n, y_n)$, with $x_1 < \cdots < x_n$, we wish to describe them as a *waveform*, that is a concatenation of k acceptable *segments* $f_1, \ldots, f_k$. Each segment f_t, $1 \leq t \leq k$ is a member of a parametric family of functions, (in our case polynomials up to a certain maximal degree, each polynomial being parameterized by its degree and coefficients) and is associated with the interval $[x_{n_{t-1}+1}, x_{n_t}]$, $n_0 = 0 < n_1 < \cdots < n_k = n$. A segment is *acceptable* if it approximates, in some norm, the data points in the associated interval.

The algorithm given by Keren *et al.* [14] is based on the Rissanen's minimum description length principle [24]: of all the acceptable waveforms, choose that with the shortest description. The description of each segment consists of its length (interval size), degree and coefficients, i.e., the description length of a constant is 3, a linear function is 4, a quadratic is 5, etc. The description length of a waveform is the sum of description lengths of its segments. The digraph $G = (V, E)$ is defined with V being the set $\{1, \ldots, n, n+1\}$ and E containing $(i, j+1)$, $1 \leq i < j \leq n$ if the segment associated with $[x_i, x_j]$ is acceptable. In this case the arc $(i, j+1)$ is weighted by the description length of the segment. Finding the waveform with minimum description length is then equivalent to finding the minimum weight path in G from 1 to $n+1$.

Keren *at el.* use least squares regression to fit a segment to the interval $[x_i, x_j]$, first fitting a constant, then a line, then a quadratic, etc., stopping with the first acceptable fit. The fit is considered acceptable if the χ^2 distribution of the residuals is less than a certain threshold. We have robustified the algorithm [32] by replacing the least squares fit with a least median of squares fit and checking the acceptability of the fit on the inliers only. An advantage of the Keren, Marcus and Werman algorithm is that it only takes $O(n^2)$ time to find the best fitting segment and to check its acceptability for all of the $n(n-1)/2$ intervals. We have used heuristic speedups of the least median of squares algorithm to achieve a time complexity of $O(n^3)$. An addition heuristic argument, enables a speedup to perhaps $O(n^2)$, and only for certain types of data was the final waveform affected.

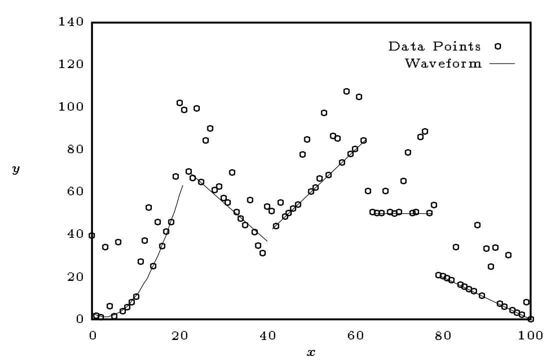

Figure 4: Least median of squares based minimum description length segmentation.

Figure 4 shows an example of our algorithm. The data points were generated by the function

$$f(x) = \begin{cases} 0.2x^2 - 1.2x + 2.5 & x < 19 \\ -0.1x^2 + 4x + 27 & 19 \le x < 41 \\ 2x - 40 & 41 \le x < 63 \\ 50 & 63 \le x < 79 \\ -x + 100 & x \ge 79 \end{cases}$$

for $x = 0, 1, \ldots, 100$. Three fifth's of the points were contaminated by a Gaussian distribution with $\sigma = 0.1$ and the remaining 40% (the outliers) were contaminated by a uniform noise in $[0, 40]$ which is not symmetric. The waveform found by our algorithm is

$$f(x) = \begin{cases} 0.20x^2 - 1.21x + 3.17 & x = 0, \ldots, 21 \\ -1.84x + 110.57 & x = 22, \ldots, 40 \\ 2.00x - 39.80 & x = 41, \ldots, 63 \\ 50.03 & x = 64, \ldots, 78 \\ -1.00x + 99.93 & x = 79, \ldots, 100, \end{cases}$$

which is also shown in Figure 4. This is a good approximation considering the large amount of contamination. The main problem is that the second segment is linear and not quadratic.

Keren, Marcus and Werman show how the one dimensional algorithm presented above should be adapted for two dimensional images and use this adaptation to segment and compress images. Our algorithm can be used in these adaptations without change.

Robust Quick Search

Often it is necessary to search an image quickly for the location of certain shapes. This allows concentration of future processing power in a smaller subimage. We have performed this task be tessellating the image into large squares and performing a least median of squares fit for the shape in each square of the tessellation. A successful fit, that is one with a small median of absolute deviations, shows the existence and location of the shape. A high breakdown point fit is necessary because background image pixels, pixels belonging to other shapes as well as genuine noise are all treated as noise.

Increasing the size of the squares decreases the speed of the search but increases the risk of not finding an instance of the sought shape. Decreasing the size of the squares increases the speed of the search, and for very small squares the shape may not be recognizable. (A small patch of any smooth surface looks like a piece of a plane.)

The above figures show examples of this search technique on depth images, with black pixels closest to the camera and white pixels furthest. In this case we were searching for spheres, parameterized explicitly as $z(x, y) = c \pm \sqrt{r^2 - (x-a)^2 - (y-b)^2}$. Figure 5 shows a 256×256 image containing four spheres on a planar background. Figure 6 shows the image with 40% of the pixels replaced by random values (uniformly generated in $\{0, \ldots, 255\}$) on the left and the same image with 40% of the pixels blacked out on the right. Table 2 lists the parameters of the four spheres, the x-coordinate going from left to right the y-coordinate

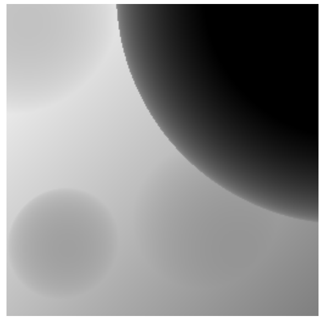

Figure 5: A depth image of four spheres.

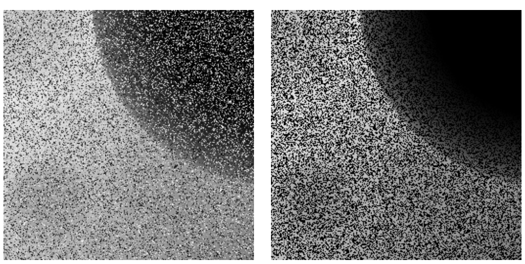

Figure 6: Figure 5 with 40% of the pixels replaced by uniform noise (left) or blacked out (right).

Sphere	Center	Radius
1	(17.5,15,290)	95.80
2	(190,200.5,280)	130.90
3	(280,-10,150)	190.50
4	(50,200,210.5)	50.00

Table 2: The spheres of Figure 5

.

Figure 5								Figure 6							
1	1	1	3	3				1	1		3	3			
1	1		3	3	3			1	1		3	3			
1			3	3						3	3				
				3	3	3	3						3	3	3
				2	3	3	3					2		3	3
4	4	4	2	2	2	2	3		4			2	2		3
4	4	4		2	2	2		4	4			2	2		
	4														

Table 3: The results with a 32×32 square.

Figure 5				Figure 6	
1	3	3		1	
		3	3		3
4		2	3		
4		2			

Table 4: The results with a 64×64 square.

from *top to bottom* and the z-coordinate indicating the depth. Some of the spheres are only partly visible, being obscured by other spheres or disappearing behind the planar background. Both the spheres numbered 1 and 2 have much smaller apparent radii owing to the obscuring by the plan. In addition sphere number 2 is obscured by sphere number 3.

Table 3 contains the results of the algorithm using a 32×32 square. In all cases the correct spheres were found. The table marks the number of each sphere found in a particular square. The reason that the upper left corner is not marked with 3's is that the big sphere is almost flat there and looks too much like a plane. Table 4 contains the results of the algorithm using a 64×64 square. This tessellation is too large to find the spheres 2 and 4 in the noisy images.

References

[1] Paul J. Besl, Jeffrey B. Birch, and Layne T. Watson. Robust window operators. In *Second International Conference on Computer Vision*, pages 591–600, Tampa, FL, December 1988.

[2] Fred L. Bookstein. Fitting conic sections to scattered data. *Computer Graphics and Image Processing*, 9(1):56–71, January 1979.

[3] Richard Cole, Jeffrey S. Salowe, W. L. Steiger, and Endre Szemerédi. An optimal-time algorithm for slope selection. *SIAM Journal on Computing*, 18(4):792–810, August 1989.

[4] David L. Donoho and Peter J. Huber. The notion of breakdown point. In Peter J. Bickel, Kjell A. Doksum, and J. L. Hodges, Jr., editors, *A Festschrift for Erich L. Lehman*, pages 157–184. Wadsworth International, 1983.

[5] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley, 1973.

[6] Frank R. Hampel, Elvezio M. Ronchetti, Peter J. Rousseeuw, and Werner A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley, 1986.

[7] Robert M. Haralick and Hyonam Joo. 2d–3d pose estimation. In *Ninth International Conference on Pattern Recognition, Volume I*, pages 385–391, Rome, Italy, November 1988.

[8] Robert M. Haralick, Hyonam Joo, Chung-Nan Lee, Xinhua Zhuang, Vinay G. Vaidya, and Man Bae Kim. Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1426–1446, November/December 1989.

[9] David C. Hoaglin, Frederick Mosteller, and John W. Tukey, editors. *Understanding Robust and Exploratory Data Analysis*. John Wiley, 1983.

[10] David C. Hoaglin, Frederick Mosteller, and John W. Tukey, editors. *Exploring Data Tables, Trends and Shapes.* John Wiley, 1985.

[11] Peter J. Huber. *Robust Statistics*. John Wiley, 1981.

[12] Boris Iglewicz. Robust scale estimators and confidence intervals for location. In David C. Hoaglin, Frederick Mosteller, and John W. Tukey, editors, *Understanding Robust and Exploratory Data Analysis*, chapter 12. John Wiley, 1983.

[13] Behzad Kamgar-Parsi, Behrooz Kamgar-Parsi, and Nathan S. Netanyahu. A nonparametric method for fitting a straight line to a noisy image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(9):998–1001, September 1989.

[14] Daniel Keren, Ruth Marcus, and Michael Werman. Segmenting and compressing waveforms by mimimum length encoding. In *Sixth Israeli Conference on Artificial Intelligence and Computer Vision*, pages 379–389, Ramat Gan, Israel, December 1989.

[15] Dong Yoon Kim, John J. Kim, Peter Meer, Doron Mintz, and Azriel Rosenfeld. Robust computer vision: A least median of squares based approach. In *DARPA Image Understanding Workshop*, pages 1117–1134, Palo Alto, CA, May 1989.

[16] Rakesh Kumar and Alllen R. Hanson. Robust estimation of camera location and orientation from noisy data having outliers. In *Workshop on Interpretation of 3D Scenes*, pages 52–60, Austin, TX, November 1989.

[17] Rakesh Kumar and Alllen R. Hanson. Analysis of different robust methods for pose refinement. In *First International Workshop on Robust Computer Vision*, pages 167–182, Seattle, WA, October 1990.

[18] Guoying Li. Robust regression. In David C. Hoaglin, Frederick Mosteller, and John W. Tukey, editors, *Exploring Data Tables, Trends and Shapes*, chapter 8. John Wiley, 1985.

[19] Kantilal Varichand Mardia. *Statistics of Directional Data*. Academic Press, 1972.

[20] Peter Meer and Doron Mintz. Robust estimators in computer vision: An introduction to least median of squares regression. In *Seventh Israeli Symposium on Artificial Intelligence and Computer Vision*, Ramat Gan, Israel, December 1990.

[21] Peter Meer, Doron Mintz, and Azriel Rosenfeld. Least median of squares based robust analysis of image structure. Technical Report CAR-TR-490, Computer Vision Laboratory, University of Maryland, College Park, MD, March 1990.

[22] Peter Meer, Doron Mintz, and Azriel Rosenfeld. Robust recovery of piecewise polynomial image structure. In *First International Workshop on Robust Computer Vision*, pages 109–126, Seattle, WA, October 1990.

[23] Vishvjit S. Nalwa and Eric Pauchon. Edgel aggregation and edge discription. *Computer Vision, Graphics and Image Processing*, 40(1):79–94, October 1987.

[24] Jorma Rissanen. Minimum discription length principle. In *Encyclopedia of Statistical Sciences*, volume 5, pages 523–527. John Wiley, 1985.

[25] Gerhard Roth and Martin D. Levine. Random sampling for primitive extraction. In *First International Workshop on Robust Computer Vision*, pages 352–366, Seattle, WA, October 1990.

[26] Gerhard Roth and Martin D. Levine. Segmentation of geometric signals using robust fitting. In *Tenth International Conference on Pattern Recognition, Volume I*, pages 826–831, Atlantic City, NJ, June 1990.

[27] Peter J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, December 1984.

[28] Peter J. Rousseeuw and Annick M. Leroy. *Robust Regression and Outlier Detection*. John Wiley, 1987.

[29] Paul D. Sampson. Fitting conic sections to "very scattered" data: An iterative refinement of the Bookstein algorithm. *Computer Graphics and Image Processing*, 18(1):97–108, January 1982.

[30] Andrew F. Siegel. Robust regression using repeated medians. *Biometrika*, 69(1):242–244, 1982.

[31] Diane L. Souvaine and J. Michael Steele. Time- and space-efficient algorithms for least median of squares regression. *Journal of the American Statistical Association*, 82(399):794–801, September 1987.

[32] Andrew Stein. Robust statistics in computer vision. Master's thesis, Hebrew University, Jerusalem, Israel, April 1991.

[33] Robert L. Stevenson and Edward J. Delp. Fitting curves with discontinuities. In *First International Workshop on Robust Computer Vision*, pages 127–136, Seattle, WA, October 1990.

[34] H. Theil. A rank-invariant method of linear and polynomial regression analysis (parts 1–3). *Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen, Ser. A*, 53:386–392, 521–525 & 1397–1412, 1950.

[35] Isaac Weiss. Line fitting in a noisy image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(3):325–329, March 1989.