# Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting

Zhengyou Zhang

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# *Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting*

Zhengyou Zhang

**N° 2676**

Octobre 1995

PROGRAMME 4


ISSN 0249-6399

# Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting

Zhengyou Zhang

Programme 4 — Robotique, image et vision
Projet Robotvis

Rapport de recherche n ° 2676 — Octobre 1995 — 44 pages

**Abstract:** Almost all problems in computer vision are related in one form or another to the problem of estimating parameters from noisy data. In this tutorial, we present what is probably the most commonly used techniques for parameter estimation. These include linear least-squares (pseudo-inverse and eigen analysis); orthogonal least-squares; gradient-weighted least-squares; bias-corrected renormalization; Kalman filtering; and robust techniques (clustering, regression diagnostics, M-estimators, least median of squares). Particular attention has been devoted to discussions about the choice of appropriate minimization criteria and the robustness of the different techniques. Their application to conic fitting is described.

**Key-words:** Parameter estimation, Least-squares, Bias correction, Kalman filtering, Robust regression

*(Résumé : tsvp)*

Unité de recherche INRIA Sophia-Antipolis
2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex (France)
Téléphone : (33) 93 65 77 77 – Télécopie : (33) 93 65 77 65

# Techniques d'estimation de paramètres : un *tutorial* avec application à l'ajustement de coniques

**Résumé :** Presque tous les problèmes en vision par ordinateur sont reliés d'une manière ou l'autre au problème de l'estimation de paramètres à partir de données bruitées. Dans ce *tutorial*, nous présentons des techniques qui sont probablement les plus utilisées pour l'estimation de paramètres. Ce sont la technique des moindre-carrés linéaires (pseudo-inverse, analyse par vecteurs propres), la régression orthogonale, la re-normalisation avec correction de biais, le filtrage de Kalman, et des techniques robustes (*clustering*, régression par diagnostics, M-estimateurs, la moindre médiane des carrés). Un effort particulier est consacré aux discussions sur le choix d'un critère de minimisation approprié et sur la robustesse des différentes techniques. Le problème de l'ajustement de coniques est utilisé pour illustrer l'exposé.

**Mots-clé :** Estimation de paramètres, moindre-carrés, correction de biais, filtrage de Kalman, régression robuste

# Contents

# 1 Introduction

Almost all problems in computer vision are related in one form or another to the problem of estimating parameters from noisy data. A few examples are line fitting, camera calibration, image matching, surface reconstruction, pose determination, and motion analysis. A parameter estimation problem is usually formulated as an optimization one. Because of different optimization criteria and because of several possible parameterizations, a given problem can be solved in many ways. The purpose of this paper is to show the importance of choosing an appropriate criterion. This will influence the accuracy of the estimated parameters, the efficiency of computation, the robustness to predictable or unpredictable errors. Conic fitting is used to illustrate these aspects because:

- it is one of the simplest problems in computer vision on one hand;
- it is, on the other hand, a relatively difficult problem because of its nonlinear nature.

Needless to say the importance of conics in our daily life and in industry.

# 2 A Glance over Parameter Estimation in General

Parameter estimation is a discipline that provides tools for the efficient use of data for aiding in mathematically modelling of phenomena and the estimation of constants appearing in these models [2]. It can thus be visualized as a study of inverse problems. Much of parameter estimation can be related to four optimization problems:

- **criterion:** the choice of the best function to optimize (minimize or maximize)
- **estimation:** the optimization of the chosen function
- **design:** optimal design to obtain the best parameter estimates
- **modeling:** the determination of the mathematical model which best describes the system from which data are measured.

In this paper we are mainly concerned with the first three problems, and we assume the model is known (a conic in the examples).

Let $\mathbf{p}$ be the (state/parameter) vector containing the parameters to be estimated. The dimension of $\mathbf{p}$, say $m$, is the number of parameters to be estimated. Let $\mathbf{z}$ be

the (measurement) vector which is the output of the system to be modeled. The system is described by a vector function $\mathbf{f}$ which relates $\mathbf{z}$ to $\mathbf{p}$ such that

$$\mathbf{f}(\mathbf{p}, \mathbf{z}) = 0 \, .$$

In practice, observed measurements $\mathbf{y}$ are only available for the system output $\mathbf{z}$ corrupted with noise $\epsilon$, i.e.,

$$\mathbf{y} = \mathbf{z} + \epsilon \, .$$

We usually make a number of measurements for the system, say $\{\mathbf{y}_i\}$ $(i = 1, \ldots, n)$, and we want to estimate $\mathbf{p}$ using $\{\mathbf{y}_i\}$. As the data are noisy, the function $\mathbf{f}(\mathbf{p}, \mathbf{y}_i) = 0$ is not valid anymore. In this case, we write down a function

$$\mathcal{F}(\mathbf{p}, \ \mathbf{y}_1, \ldots, \mathbf{y}_n)$$

which is to be optimized (without loss of generality, we will minimize the function). This function is usually called *the cost function* or *the objective function*.

If there are no constraints on $\mathbf{p}$ and the function $\mathcal{F}$ has first and second partial derivatives everywhere, necessary conditions for a minimum are

$$\frac{\partial \mathcal{F}}{\partial \mathbf{p}} = 0$$

and

$$\frac{\partial^2 \mathcal{F}}{\partial \mathbf{p}^2} > 0 \, .$$

By the last, we mean that the $m \times m$-matrix is positive definite.

# 3 Conic Fitting Problem

The problem is to fit a conic section to a set of $n$ points $\{\mathbf{x}_i\} = \{(x_i, y_i)\}$ $(i = 1, \ldots, n)$. A conic can be described by the following equation:

$$Q(x, y) = Ax^2 + 2Bxy + Cy^2 + 2Dx + 2Ey + F = 0 \, , \tag{1}$$

where $A$ and $C$ are not simultaneously zero. In practice, we encounter ellipses, where we must impose the constraint $B^2 - AC < 0$. However, this constraint is usually ignored during the fitting because

- the constraint is usually satisfied if data are not all situated in a flat section.
- the computation will be very expensive if this constraint is considered.

As the data are noisy, it is unlikely to find a set of parameters $(A, B, C, D, E, F)$ (except for the trivial solution $A = B = C = D = E = F = 0$) such that $Q(x_i, y_i) = 0$, $\forall i$. Instead, we will try to estimate them by minimizing some objective function.

# 4 Least-Squares Fitting Based on Algebraic Distances

As said before, for noisy data, the system equation, $Q(x, y) = 0$ in the case of conic fitting, can hardly hold true. A common practice is to directly minimize the *algebraic distance* $Q(x_i, y_i)$, i.e., to minimize the following function:

$$\mathcal{F} = \sum_{i=1}^{n} Q^2(x_i, y_i) \ .$$

Clearly, there exists a trivial solution $A = B = C = D = E = F = 0$. In order to avoid it, we should normalize $Q(x, y)$. There are many different normalizations proposed in the literature. Here we describe three of them.

## 4.1 Normalization with $A + C = 1$

Since the trace $A + C$ can never be zero for an ellipse, the arbitrary scale factor in the coefficients of the conic equation can be naturally removed by the normalization $A + C = 1$. This normalization has been used by many researchers [16, 20]. All ellipse can then be described by a 5-vector

$$\mathbf{p} = [A, B, D, E, F]^T \ ,$$

and the system equation $Q(x_i, y_i) = 0$ becomes:

$$\mathbf{f}_i \stackrel{\triangle}{=} \mathbf{a}_i^T \mathbf{p} - b_i = 0 \ , \tag{2}$$

where

$$\mathbf{a}_i = [x_i^2 - y_i^2, 2x_i y_i, 2x_i, 2y_i, 1]^T$$

$$b_i = -y_i^2 \ .$$

Given $n$ points, we have the following vector equation:

$$\mathbf{Ap} = \mathbf{b} \ ,$$

where

$$\begin{aligned} \mathbf{A} &= [\mathbf{a}_1, \mathbf{a}_n, \ldots, \mathbf{a}_n]^T \\ \mathbf{b} &= [b_1, b_2, \ldots, b_n]^T \ . \end{aligned}$$

The function to minimize becomes

$$\mathcal{F}(\mathbf{p}) = (\mathbf{Ap} - \mathbf{b})^T(\mathbf{Ap} - \mathbf{b}) \ .$$

Obtaining its partial derivative with respect to $\mathbf{p}$ and setting it to zero yield:

$$2\mathbf{A}^T(\mathbf{Ap} - \mathbf{b}) = \mathbf{0} \ .$$

The solution is readily given by

$$\mathbf{p} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \ . \tag{3}$$

This method is known as *pseudo inverse technique*.

### 4.2 Normalization with $A^2 + B^2 + C^2 + D^2 + E^2 + F^2 = 1$

Let $\mathbf{p} = [A, B, C, D, E, F]^T$. As $\|\mathbf{p}\|^2$, i.e., the sum of squared coefficients, can never be zero for a conic, we can set $\|\mathbf{p}\| = 1$ to remove the arbitrary scale factor in the conic equation. The system equation $Q(x_i, y_i) = 0$ becomes

$$\mathbf{a}_i^T\mathbf{p} = 0 \quad \text{with } \|\mathbf{p}\| = 1 \ ,$$

where $\mathbf{a}_i = [x_i^2, 2x_iy_i, y_i^2, 2x_i, 2y_i, 1]^T$.

Given $n$ points, we have the following vector equation:

$$\mathbf{Ap} = \mathbf{0} \quad \text{with } \|\mathbf{p}\| = 1 \ ,$$

where $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]^T$. The function to minimize becomes:

$$\mathcal{F}(\mathbf{p}) = (\mathbf{Ap})^T(\mathbf{Ap}) \stackrel{\triangle}{=} \mathbf{p}^T\mathbf{Bp} \quad \text{subject to } \|\mathbf{p}\| = 1 \ , \tag{4}$$

where $\mathbf{B} = \mathbf{A}^T\mathbf{A}$ is a symmetric matrix. The solution is the eigenvector of $\mathbf{B}$ corresponding to the smallest eigenvalue (see below).

Indeed, any $m \times m$ symmetric matrix $\mathbf{B}$ ($m = 6$ in our case) can be decomposed as

$$\mathbf{B} = \mathbf{U}\mathbf{E}\mathbf{U}^T \ , \tag{5}$$

with

$$\mathbf{E} = \mathrm{diag}(v_1, v_2, \ldots, v_m) \quad \text{and} \quad \mathbf{U} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m] \ ,$$

where $v_i$ is the $i$-th eigenvalue, and $\mathbf{e}_i$ is the corresponding eigenvector. Without loss of generality, we assume $v_1 \leq v_2 \leq \cdots \leq v_m$. The original problem (4) can now be restated as:

*Find* $a_1, a_2, \ldots, a_m$ *such that* $\mathbf{p}^T\mathbf{B}\mathbf{p}$ *is minimized with* $\mathbf{p} = a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + \cdots + a_m\mathbf{e}_m$ *subject to* $a_1^2 + a_2^2 + \cdots + a_m^2 = 1$.

After some simple algebra, we have

$$\mathbf{p}^T\mathbf{B}\mathbf{p} = a_1^2 v_1 + a_2^2 v_2 + \cdots + a_m^2 v_m \ .$$

The problem now becomes to minimize the following unconstrained function:

$$\mathcal{J} = a_1^2 v_1 + a_2^2 v_2 + \cdots + a_m^2 v_m + \lambda(a_1^2 + a_2^2 + \cdots + a_m^2 - 1) \ ,$$

where $\lambda$ is the Lagrange multiplier. Setting the derivatives of $J$ with respect to $a_1$ through $a_m$ and $\lambda$ yields:

$$\begin{aligned}
&2a_1 v_1 + 2a_1\lambda = 0 \\
&2a_2 v_2 + 2a_2\lambda = 0 \\
&\cdots\cdots \\
&2a_m v_m + 2a_m\lambda = 0 \\
&a_1^2 + a_2^2 + \cdots + a_m^2 - 1 = 0
\end{aligned}$$

There exist $m$ solutions. The $i$-th solution is given by

$$a_i = 1, \quad a_j = 0 \text{ for } j = 1, \ldots, m, \text{ except } i, \quad \lambda = -v_i \ .$$

The value of $\mathcal{J}$ corresponding to the $i$-th solution is

$$\mathcal{J}_i = v_i \ .$$

Since $v_1 \leq v_2 \leq \cdots \leq v_m$, the first solution is the one we need (the least-squares solution), i.e.,

$$a_1 = 1, \quad a_j = 0 \text{ for } j = 2, \ldots, m.$$

Thus the solution to the original problem (4) is the eigenvector of $\mathbf{B}$ corresponding to the smallest eigenvalue.

### 4.3 Normalization with $F = 1$

Another commonly used normalization is to set $F = 1$. If we use the same notations as in the last subsection, the problem becomes to minimize the following function:

$$\mathcal{F}(\mathbf{p}) = (\mathbf{Ap})^T(\mathbf{Ap}) = \mathbf{p}^T\mathbf{Bp} \quad \text{subject to } p_6 = 1 \,, \tag{6}$$

where $p_6$ is the sixth element of vector $\mathbf{p}$, i.e., $p_6 = F$.

Indeed, we seek for a least-squares solution to $\mathbf{Ap} = \mathbf{0}$ under the constraint $p_6 = 1$. The equation can be rewritten as

$$\mathbf{A}'\mathbf{p}' = -\mathbf{a}_n \,,$$

where $\mathbf{A}'$ is the matrix formed by the first $(n-1)$ columns of $\mathbf{A}$, $\mathbf{a}_n$ is the last column of $\mathbf{A}$ and $\mathbf{p}'$ is the vector $[A, B, C, D, E]^T$. The problem can now be solved by the technique described in Sect. 4.1.

In the following, we present another technique for solving this kind of problems, i.e.,

$$\mathbf{Ap} = 0 \quad \text{subject to } p_m = 1$$

based on eigen analysis, where we consider a general formulation, that is $\mathbf{A}$ is a $n \times m$ matrix, $\mathbf{p}$ is a $m$-vector, and $p_m$ is the last element of vector $\mathbf{p}$. The function to minimize is

$$\mathcal{F}(\mathbf{p}) = (\mathbf{Ap})^T(\mathbf{Ap}) \stackrel{\triangle}{=} \mathbf{p}^T\mathbf{Bp} \quad \text{subject to } p_m = 1 \,. \tag{7}$$

As in the last subsection, the symmetric matrix $\mathbf{B}$ can be decomposed as in (5), i.e., $\mathbf{B} = \mathbf{UEU}^T$. Now if we normalize each eigenvalue and eigenvector by the last element of the eigenvector, i.e.,

$$v_i' = v_i e_{im}^2 \qquad \mathbf{e}_i' = \frac{1}{e_{im}}\mathbf{e}_i \,,$$

where $e_{im}$ is the last element of the eigenvector $\mathbf{e}_i$, then the last element of the new eigenvector $\mathbf{e}_i'$ is equal to one. We now have

$$\mathbf{B} = \mathbf{U}'\mathbf{E}'\mathbf{U}'^T \,,$$

where $\mathbf{E}' = \text{diag}(v_1', v_2', \ldots, v_m')$ and $\mathbf{U}' = [\mathbf{e}_1', \mathbf{e}_2', \ldots, \mathbf{e}_m']$. The original problem (7) becomes:

*Find* $a_1, a_2, \ldots, a_m$ *such that* $\mathbf{p}^T\mathbf{B}\mathbf{p}$ *is minimized with* $\mathbf{p} = a_1\mathbf{e}'_1 + a_2\mathbf{e}'_2 + \cdots + a_m\mathbf{e}'_m$ *subject to* $a_1 + a_2 + \cdots + a_m = 1$.

After some simple algebra, we have

$$\mathbf{p}^T\mathbf{B}\mathbf{p} = a_1^2 v'_1 + a_2^2 v'_2 + \cdots + a_m^2 v'_m \ .$$

The problem now becomes to minimize the following unconstrained function:

$$\mathcal{J} = a_1^2 v'_1 + a_2^2 v'_2 + \cdots + a_m^2 v'_m + \lambda(a_1 + a_2 + \cdots + a_m - 1) \ ,$$

where $\lambda$ is the Lagrange multiplier. Setting the derivatives of $\mathcal{J}$ with respect to $a_1$ through $a_m$ and $\lambda$ yields:

$$\begin{aligned}
&2a_1 v'_1 + \lambda = 0 \\
&2a_2 v'_2 + \lambda = 0 \\
&\cdots\cdots \\
&2a_m v'_m + \lambda = 0 \\
&a_1 + a_2 + \cdots + a_m - 1 = 0
\end{aligned}$$

The unique solution to the above equations is given by

$$a_i = \frac{1}{v'_i} S \quad \text{for } i = 1, \ldots, m \ ,$$

where $S = 1 \Big/ \sum_{j=1}^{m} \frac{1}{v'_j}$. The solution to the problem (7) is given by

$$\mathbf{p} = \sum_{i=1}^{m} a_i \mathbf{e}'_i = \left( \sum_{i=1}^{m} \frac{1}{v'_j} \mathbf{e}'_i \right) \Big/ \left( \sum_{j=1}^{m} \frac{1}{v'_j} \right) \ .$$

Note that this normalization ($F = 1$) has singularities for all conics going through the origin. That is, this method cannot fit such conics because they require to set $F = 0$. This might suggest that the other normalizations are superior to the $F = 1$ normalization with respect to singularities. However, as shown in [19], the singularity problem can be overcome by shifting the data so that they are centered on the origin, and better results by setting $F = 1$ has been obtained than by setting $A + C = 1$.

# 5 Least-Squares Fitting Based on Euclidean Distances

In the above section, we have described three general techniques for solving linear least-squares problems, either unconstrained or constrained, based on algebraic distances. In this section, we describe why such techniques usually do not provide satisfactory results, and then propose to fit conics using directly Euclidean distances.

## 5.1 Why are algebraic distances usually not satisfactory ?

The big advantage of use of algebraic distances is the gain in computational efficiency, because closed-form solutions can usually be obtained. In general, however, the results are not satisfactory. There are at least two major reasons.

- The function to minimize is usually not invariant under Euclidean transformations. For example, the function with normalization $F = 1$ is not invariant with respect to translations. This is a feature we dislike, because we usually do not know in practice where is the best coordinate system to represent the data.
- A point may contribute differently to the parameter estimation depending on its position on the conic. If a priori all points are corrupted by the same amount of noise, it is desirable for them to contribute the same way. (The problem with data points corrupted by different noise will be addressed in section 8.)

To understand the second point, consider a conic in the normalized system (see Fig. 1):

$$Q(x, y) = Ax^2 + Cy^2 + F = 0 \; .$$

The algebraic distance of a point $(x_i, y_i)$ to the conic $Q$ is given by [cite Bookstein]:

$$Q(x_i, y_i) = Ax_i^2 + Cy_i^2 + F = -F(d_i^2/c_i^2 - 1) \; ,$$

where $d_i$ is the distance from the point $(x_i, y_i)$ to the center $O$ of the conic, and $c_i$ is the distance from the conic to its center along the ray from the center to the point $(x_i, y_i)$. It is thus clear that a point at the high curvature sections contributes less to the conic fitting than a point having the same amount of noise but at the low curvature sections. This is because a point at the high curvature sections has a large $c_i$ and its $|Q(x_i, y_i)|^2$ is small, while a point at the low curvature sections has a small

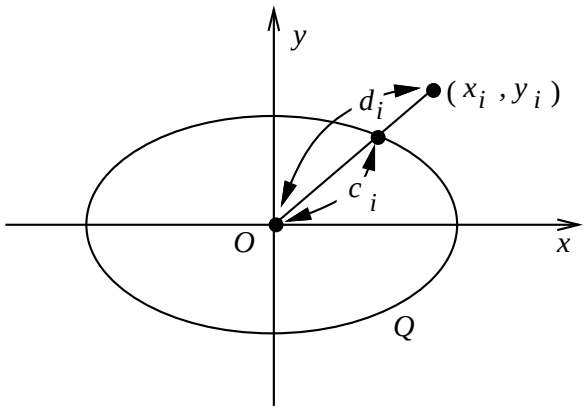


Fig. 1: Normalized conic

$c_i$ and its $|Q(x_i, y_i)|^2$ is higher with respect to the same amount of noise in the data points. Concretely, methods based on algebraic distances tend to fit better a conic to the points at low curvature sections than to those at high curvature sections. This problem is usually termed as *high curvature bias*.

### 5.2 Orthogonal distance fitting

To overcome the problems with the algebraic distances, it is natural to replace them by the orthogonal distances which are invariant to transformations in Euclidean space and which do not exhibit the high curvature bias.

The orthogonal distance $d_i$ between a point $\mathbf{x}_i = (x_i, y_i)$ and a conic $Q(x, y)$ is the Euclidean distance between $\mathbf{x}_i$ and the point $\mathbf{x}_t = (x_{ti}, y_{ti})$ in the conic whose tangent is orthogonal to the line joining $\mathbf{x}_i$ and $\mathbf{x}_t$ (see Fig. 2). Given $n$ points $\mathbf{x}_i$ $(i = 1, \ldots, n)$, the orthogonal distance fitting is to estimate the conic $Q$ by minimizing the following function

$$\mathcal{F}(\mathbf{p}) = \sum_{i=1}^{n} d_i^2 \,. \tag{8}$$

However, as the expression of $d_i$ is very complicated (see below), an iterative optimization procedure must be carried out. Many techniques are readily available, including Gauss-Newton algorithm, Steepest Gradient Descent, Levenberg-Marquardt

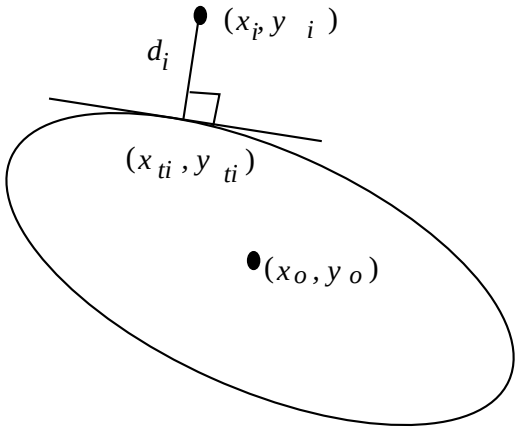


Fig. 2: Orthogonal distance of a point to a conic

procedure, and simplex method. A software `ODRPACK` (written in Fortran) for weighted orthogonal distance regression is domain public and is available from `NETLIB` (netlib@ornl.gov). Initial guess of the conic parameters must be supplied, which can be obtained using the techniques described in the last section.

Let us now proceed to compute the orthogonal distance $d_i$. The subscript $i$ will be omitted for clarity. Refer again to Fig. 2. The conic is assumed to be described by

$$Q(x, y) = A(x - x_o)^2 + B(x - x_o)(y - y_o) + C(y - y_o)^2 - 1 = 0 \, .$$

Point $\mathbf{x}_t = (x_t, y_t)$ must satisfy the following two equations:

$$A(x_t - x_o)^2 + B(x_t - x_o)(y_t - y_o) + C(y_t - y_o)^2 - 1 = 0 \tag{9}$$

$$(y - y_t)\frac{\partial}{\partial x}Q(x_t, y_t) = (x - x_t)\frac{\partial}{\partial y}Q(x_t, y_t) \tag{10}$$

Equation (9) merely says the point $\mathbf{x}_t$ is on the conic, while (10) says that the tangent at $\mathbf{x}_t$ is orthogonal to the vector $\mathbf{x} - \mathbf{x}_t$.

Let $\Delta_x = x_t - x_o$ and $\Delta_y = y_t - y_o$. From (9),

$$\Delta_y = \frac{-B\Delta_x + \xi}{2C} \, , \tag{11}$$

where $\xi^2 = B^2\Delta_x^2 - 4C(A\Delta_x^2 - 1) = (B^2 - 4AC)\Delta_x^2 + 4C$. From (10),

$$(2A\Delta_x + B\Delta_y)(y - y_o - \Delta_y) = (2C\Delta_y + B\Delta_x)(x - x_o - \Delta_x) \; .$$

Substituting the value of $\Delta_y$ (11) in the above equation leads to the following equation:

$$e_1\Delta_x^2 + e_2\Delta_x + e_3 = (e_4\Delta_x + e_5)\xi \; , \tag{12}$$

where

$$\begin{aligned} e_0 &= B^2 - 4AC \\ e_1 &= 2Be_0 \\ e_2 &= 2Ce_0(y - y_0) \\ e_3 &= 4BC \\ e_4 &= B^2 + 4C^2 + e_0 \\ e_5 &= 2C[B(y - y_0) - 2C(x - x_0)] \; . \end{aligned}$$

Squaring the above equation, we have

$$(e_1\Delta_x^2 + e_2\Delta_x + e_3)^2 = (e_4\Delta_x + e_5)^2(e_0\Delta_x^2 + 4C) \; .$$

Rearranging the terms, we obtain an equation of degree four in $\Delta_x$:

$$f_4\Delta_x^4 + f_3\Delta_x^3 + f_2\Delta_x^2 + f_1\Delta_x + f_0 = 0 \; , \tag{13}$$

where

$$\begin{aligned} f_4 &= e_1^2 - e_0e_4^2 \\ f_3 &= 2e_1e_2 - 2e_0e_4e_5 \\ f_2 &= e_2^2 + 2e_1e_3 - e_0e_5^2 - 4e_4^2C \\ f_1 &= 2e_2e_3 - 8e_4e_5C \\ f_0 &= e_3^2 - 4e_5^2C \; . \end{aligned}$$

The two or four real roots of (13) can be found in closed form. For one solution $\Delta_x$, we can obtain $\xi$ from (12), i.e.:

$$\xi = (e_1\Delta_x^2 + e_2\Delta_x + e_3)/(e_4\Delta_x + e_5) \; .$$

Thus, $\Delta_y$ is computed from (11). Eventually comes the orthogonal distance $d$, which is given by

$$d = \sqrt{(x - x_o - \Delta_x)^2 + (y - y_o - \Delta_y)^2} \; .$$

Note that we possibly have four solutions. Only the one which gives the smallest distance is the one we are seeking for.

# 6 Gradient Weighted Least-Squares Fitting

The least-squares method described in the last sections is usually called *ordinary least-squares estimator* (OLS). Formally, we are given $n$ equations:

$$f_i \stackrel{\triangle}{=} A_i^T \mathbf{p} - b_i = \varepsilon_i \; ,$$

where $\varepsilon_i$ is the additive error in the $i$-th equation with mean zero: $E(\varepsilon_i) = 0$, and variance $\Lambda_{\varepsilon_i} = \sigma_i^2$. Writing down in matrix form yields

$$\mathbf{A}\mathbf{p} - \mathbf{b} = \mathbf{e} \; .$$

where

$$\mathbf{A} = \begin{bmatrix} A_1^T \\ \vdots \\ A_n^T \end{bmatrix} \; , \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \; , \quad \text{and} \quad \mathbf{e} = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix} \; .$$

The OLS estimator tries to estimate $\mathbf{p}$ by minimizing the following sum of squared errors:

$$\mathcal{F} = \mathbf{e}^T\mathbf{e} = (\mathbf{A}\mathbf{p} - \mathbf{b})^T(\mathbf{A}\mathbf{p} - \mathbf{b}) \; ,$$

which gives, as we have already seen, the solution as

$$\mathbf{p} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \; .$$

It can be shown (see, e.g., [2]) that the OLS estimator produces the optimal estimate of $\mathbf{p}$, "optimal" in terms of *minimum covariance of* $\mathbf{p}$, if the errors $\varepsilon_i$ are uncorrelated (i.e., $E(\varepsilon_i\varepsilon_j) = \sigma_i^2\delta_{ij}$) and their variances are constant (i.e., $\Lambda_{\varepsilon_i} = \sigma^2 \; \forall i \in [1, \ldots, n]$).

Now let us examine whether the above assumptions are valid or not for conic fitting. Data points are provided by some signal processing algorithm such as edge

detection. It is reasonable to assume that errors are independent from one point to another, because when detecting a point we usually do not use any information from other points. It is also reasonable to assume the errors are constant for all points because we use the same algorithm for edge detection. However, we must note that we are talking about the errors in the points, *but not those in the equations* (i.e., $\varepsilon_i$). Let the error in a point $\mathbf{x}_i = [x_i, y_i]^T$ be Gaussian with mean zero and covariance $\Lambda_{\mathbf{x}} = \sigma^2 \mathbf{I}_2$, where $\mathbf{I}_2$ is the $2\times2$ identity matrix. That is, the error distribution is assumed to be isotropic in both directions ($\sigma_x^2 = \sigma_y^2 = \sigma^2$, $\sigma_{xy} = 0$). In Sect. 8, we will consider the case where each point may have different noise distribution. Refer to Eq. (2). We now compute the variance $\Lambda_{\varepsilon_i}$ of function $f_i$ from point $(x_i, y_i)$ and its uncertainty. Let $(\hat{x}_i, \hat{y}_i)$ be the true position of the point, we have certainly

$$\hat{f}_i = (\hat{x}_i^2 - \hat{y}_i^2)A + 2\hat{x}_i\hat{y}_i B + 2\hat{x}_i D + 2\hat{y}_i E + F + \hat{y}_i^2 = 0 \ .$$

We now expand $f_i$ into Taylor series at $x = \hat{x}_i$ and $y = \hat{y}_i$, i.e.,

$$f_i = \hat{f}_i + \frac{\partial f_i}{\partial x_i}(x_i - \hat{x}_i) + \frac{\partial f_i}{\partial y_i}(y_i - \hat{y}_i) + O\left((x_i - \hat{x}_i)^2\right) + O\left((y_i - \hat{y}_i)^2\right) \ .$$

Ignoring the high order terms, we can now compute the variance of $f_i$, i.e.,

$$\begin{aligned} \Lambda_{\varepsilon_i} &= E(f_i - \hat{f}_i)^2 = \left(\frac{\partial f_i}{\partial x_i}\right)^2 E(x_i - \hat{x}_i)^2 + \left(\frac{\partial f_i}{\partial y_i}\right)^2 E(y_i - \hat{y}_i)^2 \\ &= \left[\left(\frac{\partial f_i}{\partial x_i}\right)^2 + \left(\frac{\partial f_i}{\partial y_i}\right)^2\right]\sigma^2 \triangleq \nabla f_i^2 \sigma^2 \ , \end{aligned}$$

where $\nabla f_i$ is just the gradient of $f_i$ with respect to $x_i$ and $y_i$, and

$$\begin{aligned} \tfrac{\partial f_i}{\partial x_i} &= 2Ax_i + 2By_i + 2D \\ \tfrac{\partial f_i}{\partial y_i} &= -2Ay_i + 2Bx_i + 2E + 2y_i \ . \end{aligned} \tag{14}$$

It is now clear that the variance of each equation is not the same, and thus the OLS estimator does not yield an optimal solution.

In order to obtain a constant variance function, it is sufficient to devide the original function by its gradient, i.e.,

$$f_i' = f_i / \nabla f_i \ ,$$

then $f_i'$ has the constant variance $\sigma^2$. We can now try to find the parameters $\mathbf{p}$ by minimizing the following function:

$$\mathcal{F} = \sum f_i'^2 = \sum f_i^2 / \nabla f_i^2 = (\mathbf{A}\mathbf{p} - \mathbf{b})^T \mathbf{W}^{-1} (\mathbf{A}\mathbf{p} - \mathbf{b}) \ ,$$

where $\mathbf{W} = \mathrm{diag}(\nabla f_1^2, \nabla f_2^2, \ldots, \nabla f_n^2)$. This method is thus called *Gradient Weighted Least-Squares*, and the solution can be easily obtained by setting $\frac{\partial \mathcal{F}}{\partial \mathbf{p}} = 2(\mathbf{A}\mathbf{p} - \mathbf{b})^T \mathbf{W}^{-1} \mathbf{A} = 0$, which yields

$$\mathbf{p} = (\mathbf{A}^T \mathbf{W}^{-1} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W}^{-1} \mathbf{b} \; . \tag{15}$$

Note that the gradient-weighted LS is in general a nonlinear minimization problem and a closed-form solution does not exist. In the above, we gave a closed-form solution because we have ignored the dependence of $\mathbf{W}$ on $\mathbf{p}$ in computing $\frac{\partial \mathcal{F}}{\partial \mathbf{p}}$. In reality, $\mathbf{W}$ does depend on $\mathbf{p}$, as can be seen in Eq. (14). Therefore, the above solution is only an approximation. In practice, we run the following iterative procedure:

**step 1**: $k = 0$. Compute $\mathbf{p}^{(0)}$ using OLS. Eq. (3)
**step 2**: Compute the weight matrix $\mathbf{W}^{(k)}$
**step 3**: Compute $\mathbf{p}^{(k)}$ using Gradient Weighted LS. Eq. (15)
**step 4**: If $\mathbf{p}^{(k)}$ is very close to $\mathbf{p}^{(k-1)}$, then **stop**;
otherwise **go to step 2**.

In the above, the superscript $(k)$ denotes the iteration number.

# 7 Bias-Corrected Renormalization Fitting

Consider the biquadratic representation of an ellipse:

$$Ax^2 + 2Bxy + Cy^2 + 2Dx + 2Ey + F = 0 \; .$$

Given $n$ noisy points $\mathbf{x}_i = [x_i, y_i]^T$ $(i = 1, \ldots, n)$, we want to estimate the coefficients of the ellipse: $\mathbf{p} = [A,\ B,\ C,\ D,\ E,\ F]^T$. Due to the homogeneity, we set $\|\mathbf{p}\| = 1$.

For each point $\mathbf{x}_i$, we thus have one scalar equation:

$$f_i = M_i^T \mathbf{p} = 0 \; ,$$

where

$$M_i = [x_i^2,\ 2x_i y_i,\ y_i^2,\ 2x_i,\ 2y_i,\ 1]^T \; .$$

Hence, $\mathbf{p}$ can be estimated by minimizing the following objective function (weighted least-squares optimization)

$$\mathcal{F} = \Sigma_{i=1}^n w_i (M_i^T \mathbf{p})^2 \; , \tag{16}$$

where $w_i$'s are positive weights.

Assume that each point has the same error distribution with mean zero and covariance $\Lambda_{\mathbf{x}_i} = \begin{bmatrix} \sigma^2 & \\ & \sigma^2 \end{bmatrix}$. The covariance of $f_i$ is then given by

$$\Lambda_{f_i} = \begin{bmatrix} \frac{\partial f_i}{\partial x_i} & \frac{\partial f_i}{\partial x_i} \end{bmatrix} \Lambda_{\mathbf{x}_i} \begin{bmatrix} \frac{\partial f_i}{\partial x_i} & \frac{\partial f_i}{\partial x_i} \end{bmatrix}^T ,$$

where

$$\begin{array}{ll} \frac{\partial f_i}{\partial x_i} & = 2(Ax_i + By_i + D) \\ \frac{\partial f_i}{\partial y_i} & = 2(Bx_i + Cy_i + D) \ . \end{array}$$

Thus we have

$$\begin{array}{ll} \Lambda_{f_i} & = 4\sigma^2[(A^2+B^2)x_i^2 + (B^2+C^2)y_i^2 \\ & +2B(A+C)x_iy_i + 2(AD+BE)x_i + 2(BD+CE)y_i + (D^2+E^2)] \ . \end{array}$$

The weights can then be chosen to the inverse proportion of the variances. Since multiplication by a constant does not affect the result of the estimation, we set

$$\begin{array}{ll} w_i & = 4\sigma^2/\Lambda_{f_i} = 1/[(A^2+B^2)x_i^2 + (B^2+C^2)y_i^2 \\ & +2B(A+C)x_iy_i + 2(AD+BE)x_i + 2(BD+CE)y_i + (D^2+E^2)] \ . \end{array}$$

The objective function (16) can be rewritten as

$$\mathcal{F} = \mathbf{p}^T \left( \Sigma_{i=1}^n \, w_i M_i M_i^T \right) \mathbf{p} \ ,$$

which is a quadratic form in unit vector $\mathbf{p}$. Let

$$\mathbf{N} = \Sigma_{i=1}^n \, w_i M_i M_i^T \ .$$

The solution is the eigenvector of $\mathbf{N}$ associated to the smallest eigenvalue.

If each point $\mathbf{x}_i = [x_i, y_i]^T$ is perturbed by noise of $\Delta\mathbf{x}_i = [\Delta x_i, \Delta y_i]^T$ with

$$E[\Delta\mathbf{x}_i] = 0 \ , \quad \text{and} \quad \Lambda_{\Delta\mathbf{x}_i} = E[\Delta\mathbf{x}_i \Delta\mathbf{x}_i^T] = \begin{bmatrix} \sigma^2 & \\ & \sigma^2 \end{bmatrix} \ ,$$

the matrix $\mathbf{N}$ is perturbed accordingly: $\mathbf{N} = \bar{\mathbf{N}} + \Delta\mathbf{N}$, where $\bar{\mathbf{N}}$ is the unperturbed matrix. If $E[\Delta\mathbf{N}] = 0$, then the estimate is *statistically unbiased*; otherwise, it is *statistically biased*, because following the perturbation theorem the bias of $\mathbf{p}$, i.e., $E[\Delta\mathbf{p}] = O(E[\Delta\mathbf{N}])$.

Let $\mathbf{N}_i = M_i M_i^T$, then $\mathbf{N} = \Sigma_{i=1}^n w_i \mathbf{N}_i$. We have

$$\mathbf{N}_i = \begin{bmatrix} x_i^4 & 2x_i^3 y_i & x_i^2 y_i^2 & 2x_i^3 & 2x_i^2 y_i & x_i^2 \\ 2x_i^3 y_i & 4x_i^2 y_i^2 & 2x_i y_i^3 & 4x_i^2 y_i & 4x_i y_i^2 & 2x_i y_i \\ x_i^2 y_i^2 & 2x_i y_i^3 & y_i^4 & 2x_i y_i^2 & 2y_i^3 & y_i^2 \\ 2x_i^3 & 4x_i^2 y_i & 2x_i y_i^2 & 4x_i^2 & 4x_i y_i & 2x_i \\ 2x_i^2 y_i & 4x_i y_i^2 & 2y_i^3 & 4x_i y_i & 4y_i^2 & 2y_i \\ x_i^2 & 2x_i y_i & y_i^2 & 2x_i & 2y_i & 1 \end{bmatrix} .$$

If we carry out the Taylor development and ignore quantities of order higher than 2, it can be shown that the expectation of $\Delta\mathbf{N}$ is given by

$$\begin{aligned} E[\Delta\mathbf{N}] &= \sigma^2 \begin{bmatrix} 6x_i^2 & 6x_i y_i & x_i^2 + y_i^2 & 6x_i & 2y_i & 1 \\ 6x_i y_i & 4(x_i^2 + y_i^2) & 6x_i y_i & 4y_i & 4x_i & 0 \\ x_i^2 + y_i^2 & 6x_i y_i & 6y_i^2 & 2x_i & 6y_i & 1 \\ 6x_i & 4y_i & 2x_i & 4 & 0 & 0 \\ 2y_i & 4x_i & 6y_i & 0 & 4 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \\ &\triangleq c\mathcal{B}_i . \end{aligned}$$

It is clear that if we define

$$\widehat{\mathbf{N}} = \Sigma_{i=1}^n w_i \left[\mathbf{N}_i - E[\Delta\mathbf{N}]\right] = \Sigma_{i=1}^n w_i [\mathbf{N}_i - c\mathcal{B}_i] ,$$

then $\widehat{\mathbf{N}}$ is unbiased, i.e., $E[\widehat{\mathbf{N}}] = \bar{\mathbf{N}}$, and hence the unit eigenvector $\mathbf{p}$ of $\widehat{\mathbf{N}}$ associated to the smallest eigenvalue is an *unbiased* estimate of the exact solution $\bar{\mathbf{p}}$.

Ideally, the constant $c$ should be chosen so that $E[\widehat{\mathbf{N}}] = \bar{\mathbf{N}}$, but this is impossible unless image noise characteristics are known. On the other hand, if $E[\widehat{\mathbf{N}}] = \bar{\mathbf{N}}$, we have

$$E[\bar{\mathbf{p}}^T \widehat{\mathbf{N}} \bar{\mathbf{p}}] = \bar{\mathbf{p}}^T E[\widehat{\mathbf{N}}] \bar{\mathbf{p}} = \bar{\mathbf{p}}^T \bar{\mathbf{N}} \bar{\mathbf{p}} = 0 ,$$

because $\mathcal{F} = \mathbf{p}^T \mathbf{N} \mathbf{p}$ takes its aboslute minimum 0 for the exact solution $\bar{\mathbf{p}}$ in the absence of noise. This suggests that we require that $\mathbf{p}^T \mathbf{N} \mathbf{p} = 0$ at each iteration. If for the current $c$ and $\mathbf{p}$, $\mathbf{p}^T \widehat{\mathbf{N}} \mathbf{p} = \lambda_{\min} \neq 0$, we can update $c$ by $\Delta c$ such that

$$\mathbf{p}^T \Sigma_{i=1}^n [w_i \mathbf{N}_i - c w_i \mathcal{B}_i] \mathbf{p} - \mathbf{p}^T \Sigma_{i=1}^n \Delta c w_i \mathcal{B}_i \, \mathbf{p} = 0 .$$

That is,

$$\Delta c = \frac{\lambda_{\min}}{\mathbf{p}^T \Sigma_{i=1}^n w_i \mathcal{B}_i \, \mathbf{p}} .$$

To summarize, the renormalization procedure can be described as:

1. Let $c = 0$, $w_i = 1$ for $i = 1, \ldots, n$.

2. Compute the unit eigenvector $\mathbf{p}$ of

$$\widehat{\mathbf{N}} = \Sigma_{i=1}^{n}\, w_i[\mathbf{N}_i - c\mathcal{B}_i]$$

   associated to the smallest eigenvalue, which is denoted by $\lambda_{\min}$.

3. Update $c$ as

$$c \leftarrow c + \frac{\lambda_{\min}}{\mathbf{p}^T \Sigma_{i=1}^{n}\, w_i \mathcal{B}_i\, \mathbf{p}}$$

   and recompute $w_i$ using the new $\mathbf{p}$.

4. Return $\mathbf{p}$ if the update has converged; go back to step 2 otherwise.

*Remark 1:* This implementation is different from that described in the paper of Kanatani [8]. This is because in his implementation, he uses the N-vectors to represent the 2-D points. In the derivation of the bias, he assumes that the perturbation in each N-vector, i.e., $\Delta\mathbf{m}_\alpha$ in his notations, has the same magnitude $\tilde{\varepsilon} = \sqrt{E(\|\Delta\mathbf{m}_\alpha^2\|)}$. This is an unrealistic assumption. In fact, to the first order, $\Delta\mathbf{m}_\alpha = \frac{1}{\sqrt{x_\alpha^2+y_\alpha^2+f^2}} \begin{bmatrix} \Delta x_\alpha \\ \Delta y_\alpha \\ 0 \end{bmatrix}$, thus $\|\Delta\mathbf{m}_\alpha^2\|^2 = \frac{\Delta x_\alpha^2 + \Delta y_\alpha^2}{x_\alpha^2+y_\alpha^2+f^2}$. Hence, $E(\|\Delta\mathbf{m}_\alpha^2\|) = \frac{2\sigma^2}{x_\alpha^2+y_\alpha^2+f^2}$, where we assume the perturbation in image plane is the same for each point (with mean zero and standard deviation $\sigma$).

*Remark 2:* This method is optimal only in the sense of *unbiasness*. Another criterion of optimality, namely the *minimum variance of estimation*, is not addressed in this method.

# 8 Kalman Filtering Technique

**Nota:** The first four subsections are extracted from the book [24]: *3D Dynamic Scene Analysis: A Stereo Based Approach*, by Z. Zhang & O. Faugeras (Springer Berlin 1992).

Kalman filtering, as pointed out by Lowe [11], is likely to have applications throughout Computer Vision as a general method for integrating noisy measurements.

The behavior of a dynamic system can be described by the evolution of a set of variables, called *state variables*. In practice, the individual state variables of a

dynamic system cannot be determined exactly by direct measurements; instead, we usually find that the measurements that we make are functions of the state variables and that these measurements are corrupted by random noise. The system itself may also be subjected to random disturbances. It is then required to estimate the state variables from the noisy observations.

If we denote the state vector by $\mathbf{s}$ and denote the measurement vector by $\mathbf{x}'$, a dynamic system (in discrete-time form) can be described by

$$\mathbf{s}_{i+1} = \mathbf{h}_i(\mathbf{s}_i) + \mathbf{n}_i \ , \quad i = 0, 1, \cdots \ , \tag{17}$$

$$\mathbf{f}_i(\mathbf{x'}_i, \mathbf{s}_i) = 0 \ , \qquad i = 0, 1, \cdots \ , \tag{18}$$

where $\mathbf{n}_i$ is the vector of random disturbance of the dynamic system and is usually modeled as white noise:

$$E[\mathbf{n}_i] = \mathbf{0} \quad \text{and} \quad E[\mathbf{n}_i \mathbf{n}_i^T] = Q_i \ .$$

In practice, the system noise covariance $Q_i$ is usually determined on the basis of experience and intuition (i.e., it is guessed). In (18), the vector $\mathbf{x'}_i$ is called the measurement vector. In practice, the measurements that can be made contain random errors. We assume the measurement system is disturbed by additive white noise, i.e., the real observed measurement $\mathbf{x}_i$ is expressed as

$$\mathbf{x}_i = \mathbf{x'}_i + \boldsymbol{\eta}_i \ , \tag{19}$$

where

$$\begin{aligned} E[\boldsymbol{\eta}_i] &= \mathbf{0} \ , \\ E[\boldsymbol{\eta}_i \boldsymbol{\eta}_j^T] &= \begin{cases} \Lambda_{\boldsymbol{\eta}_i} & \text{for } i = j \ , \\ \mathbf{0} & \text{for } i \neq j \ . \end{cases} \end{aligned}$$

The measurement noise covariance $\Lambda_{\boldsymbol{\eta}_i}$ is either provided by some signal processing algorithm or guessed in the same manner as the system noise. In general, these noise levels are determined independently. We assume then there is no correlation between the noise process of the system and that of the observation, that is

$$E[\boldsymbol{\eta}_i \mathbf{n}_j^T] = \mathbf{0} \quad \text{for every } i \text{ and } j.$$

### 8.1 Standard Kalman Filter

When $\mathbf{h}_i(\mathbf{s}_i)$ is a linear function

$$\mathbf{s}_{i+1} = H_i \mathbf{s}_i + \mathbf{n}_i$$

and we are able to write down explicitly a linear relationship

$$\mathbf{x}_i = F_i \mathbf{s}_i + \boldsymbol{\eta}_i$$

from $\mathbf{f}_i(\mathbf{x'}_i, \mathbf{s}_i) = 0$, then the standard Kalman filter is directly applicable.

**Kalman Filter**

- Prediction of states:
  $\hat{\mathbf{s}}_{i|i-1} = H_{i-1}\hat{\mathbf{s}}_{i-1}$
- Prediction of the covariance matrix of states:
  $P_{i|i-1} = H_{i-1}P_{i-1}H_{i-1}^T + Q_{i-1}$
- Kalman gain matrix:
  $K_i = P_{i|i-1}F_i^T(F_iP_{i|i-1}F_i^T + \Lambda_{\boldsymbol{\eta}_i})^{-1}$
- Update of the state estimation:
  $\hat{\mathbf{s}}_i = \hat{\mathbf{s}}_{i|i-1} + K_i(\mathbf{x}_i - F_i\hat{\mathbf{s}}_{i|i-1})$
- Update of the covariance matrix of states:
  $P_i = (\mathbf{I} - K_iF_i)P_{i|i-1}$
- Initialization:
  $P_{0|0} = \Lambda_{\mathbf{s}_0}$
  $\hat{\mathbf{s}}_{0|0} = E[\mathbf{s}_0]$

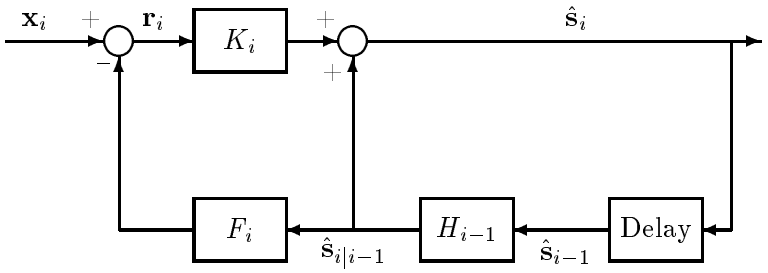


Fig. 3: Kalman filter block diagram

Figure 3 is a block diagram for the Kalman filter. At time $t_i$, the system model inherently in the filter structure generates $\hat{\mathbf{s}}_{i|i-1}$, the best prediction of the state, using the previous state estimate $\hat{\mathbf{s}}_{i-1}$. The previous state covariance matrix $P_{i-1}$ is extrapolated to the predicted state covariance matrix $P_{i|i-1}$. $P_{i|i-1}$ is then used to compute the Kalman gain matrix $K_i$ and to update the covariance matrix $P_i$. The system model generates also $F_i\hat{\mathbf{s}}_{i|i-1}$ which is the best prediction of what the measurement at time $t_i$ will be. The real measurement $\mathbf{x}_i$ is then read in, and the measurement residual (also called *innovation*)

$$\mathbf{r}_i \;=\; \mathbf{x}_i - F_i\hat{\mathbf{s}}_{i|i-1}$$

is computed. Finally, the residual $\mathbf{r}_i$ is weighted by the Kalman gain matrix $K_i$ to generate a correction term and is added to $\hat{\mathbf{s}}_{i|i-1}$ to obtain the updated state $\hat{\mathbf{s}}_i$.

The Kalman filter gives a linear, unbiased, and minimum error variance recursive algorithm to optimally estimate the unknown state of a linear dynamic system from noisy data taken at discrete real-time intervals. Without entering into the theoretical justification of the Kalman filter, for which the reader is referred to many existing books such as [7, 13], we insist here on the point that the Kalman filter yields at $t_i$ an optimal estimate of $\mathbf{s}_i$, optimal in the sense that the spread of the estimate-error probability density is minimized. In other words, the estimate $\hat{\mathbf{s}}_i$ given by the Kalman filter minimizes the following cost function

$$\mathcal{F}_i(\hat{\mathbf{s}}_i) = E[(\hat{\mathbf{s}}_i - \mathbf{s}_i)^T M (\hat{\mathbf{s}}_i - \mathbf{s}_i)] \;,$$

where $M$ is an arbitrary, positive semidefinite matrix. The optimal estimate $\hat{\mathbf{s}}_i$ of the state vector $\mathbf{s}_i$ is easily understood to be a least-squares estimate of $\mathbf{s}_i$ with the properties that [3]:

1. the transformation that yields $\hat{\mathbf{s}}_i$ from $[\mathbf{x}_0^T \;\cdots\; \mathbf{x}_i^T]^T$ is linear,
2. $\hat{\mathbf{s}}_i$ is unbiased in the sense that $E[\hat{\mathbf{s}}_i] = E[\mathbf{s}_i]$,
3. it yields a minimum variance estimate with the inverse of covariance matrix of measurement as the optimal weight.

By inspecting the Kalman filter equations, the behavior of the filter agrees with our intuition. First, let us look at the Kalman gain $K_i$. After some matrix manipulation, we express the gain matrix in the form:

$$K_i = P_i F_i^T \Lambda_{\boldsymbol{\eta}_i}^{-1} \;. \tag{20}$$

Thus, the gain matrix is "proportional" to the uncertainty in the estimate and "inversely proportional" to that in the measurement. If the measurement is very uncertain and the state estimate is relatively precise, then the residual $\mathbf{r}_i$ is resulted mainly by the noise and little change in the state estimate should be made. On the other hand, if the uncertainty in the measurement is small and that in the state estimate is big, then the residual $\mathbf{r}_i$ contains considerable information about errors in the state estimate and strong correction should be made to the state estimate. All these are exactly reflected in (20).

Now, let us examine the covariance matrix $P_i$ of the state estimate. By inverting $P_i$ and replacing $K_i$ by its explicit form (20), we obtain:

$$P_i^{-1} = P_{i|i-1}^{-1} + F_i^T \Lambda_{\boldsymbol{\eta}_i}^{-1} F_i \ . \tag{21}$$

From this equation, we observe that if a measurement is very uncertain ($\Lambda_{\boldsymbol{\eta}_i}$ is big), the covariance matrix $P_i$ will decrease only a little if this measurement is used. That is, the measurement contributes little to reducing the estimation error. On the other hand, if a measurement is very precise ($\Lambda_{\boldsymbol{\eta}_i}$ is small), the covariance $P_i$ will decrease considerably. This is logic. As described in the previous paragraph, such measurement contributes considerably to reducing the estimation error. Note that Equation (21) should not be used when measurements are noise free because $\Lambda_{\boldsymbol{\eta}_i}^{-1}$ is not defined.

## 8.2 Extended Kalman Filter

If $\mathbf{h}_i(\mathbf{s}_i)$ is not linear or a linear relationship between $\mathbf{x}_i$ and $\mathbf{s}_i$ cannot be written down, the so-called *Extended Kalman Filter* (EKF for abbreviation) can be applied[1].

The EKF approach is to apply the standard Kalman filter (for *linear* systems) to *nonlinear* systems with additive white noise by continually updating a *linearization* around the previous state estimate, starting with an initial guess. In other words, we only consider a linear Taylor approximation of the system function at the previous state estimate and that of the observation function at the corresponding predicted position. This approach gives a simple and efficient algorithm to handle a nonlinear model. However, convergence to a reasonable estimate may *not* be obtained if the initial guess is poor or if the disturbances are so large that the linearization is inadequate to describe the system.

[1] Note that in the usual formulation of the EKF, the measurement (observation) function $\mathbf{f}_i(\mathbf{x'}_i, \mathbf{s}_i)$ is of the form $\mathbf{x'}_i - \mathbf{g}_i(\mathbf{s}_i)$. Unfortunately, that formulation is not general enough to deal with the problems addressed in this monograph.

We expand $\mathbf{f}_i(\mathbf{x'}_i, \mathbf{s}_i)$ into a Taylor series about $(\mathbf{x}_i, \hat{\mathbf{s}}_{i|i-1})$:

$$\begin{aligned}\mathbf{f}_i(\mathbf{x'}_i, \mathbf{s}_i) &= \mathbf{f}_i(\mathbf{x}_i, \hat{\mathbf{s}}_{i|i-1}) + \frac{\partial \mathbf{f}_i(\mathbf{x}_i, \hat{\mathbf{s}}_{i|i-1})}{\partial \mathbf{x'}_i}(\mathbf{x'}_i - \mathbf{x}_i) + \frac{\partial \mathbf{f}_i(\mathbf{x}_i, \hat{\mathbf{s}}_{i|i-1})}{\partial \mathbf{s}_i}(\mathbf{s}_i - \hat{\mathbf{s}}_{i|i-1}) \\ &\quad + O((\mathbf{x'}_i - \mathbf{x}_i)^2) + O((\mathbf{s}_i - \hat{\mathbf{s}}_{i|i-1})^2) \ . \end{aligned} \tag{22}$$

By ignoring the second order terms, we get a linearized measurement equation:

$$\mathbf{y}_i \ = \ M_i \mathbf{s}_i \ + \ \boldsymbol{\xi}_i \ , \tag{23}$$

where $\mathbf{y}_i$ is the new measurement vector, $\boldsymbol{\xi}_i$ is the noise vector of the new measurement, and $M_i$ is the linearized transformation matrix. They are given by

$$\begin{aligned} M_i &= \frac{\partial \mathbf{f}_i(\mathbf{x}_i, \hat{\mathbf{s}}_{i|i-1})}{\partial \mathbf{s}_i} \ , \\ \mathbf{y}_i &= -\mathbf{f}_i(\mathbf{x}_i, \hat{\mathbf{s}}_{i|i-1}) + \frac{\partial \mathbf{f}_i(\mathbf{x}_i, \hat{\mathbf{s}}_{i|i-1})}{\partial \mathbf{s}_i}\hat{\mathbf{s}}_{i|i-1} \ , \\ \boldsymbol{\xi}_i &= \frac{\partial \mathbf{f}_i(\mathbf{x}_i, \hat{\mathbf{s}}_{i|i-1})}{\partial \mathbf{x'}_i}(\mathbf{x'}_i - \mathbf{x}_i) = -\frac{\partial \mathbf{f}_i(\mathbf{x}_i, \hat{\mathbf{s}}_{i|i-1})}{\partial \mathbf{x'}_i}\boldsymbol{\eta}_i \ . \end{aligned}$$

Clearly, we have $E[\boldsymbol{\xi}_i] = \mathbf{0}$, and $E[\boldsymbol{\xi}_i \boldsymbol{\xi}_i^T] = \frac{\partial \mathbf{f}_i(\mathbf{x}_i, \hat{\mathbf{s}}_{i|i-1})}{\partial \mathbf{x'}_i} \Lambda_{\boldsymbol{\eta}_i} \frac{\partial \mathbf{f}_i(\mathbf{x}_i, \hat{\mathbf{s}}_{i|i-1})}{\partial \mathbf{x'}_i}^T \ \triangleq \ \Lambda_{\boldsymbol{\xi}_i}$.

The extended Kalman filter equations are given in the following algorithm, where the derivative $\frac{\partial \mathbf{h}_i}{\partial \mathbf{s}_i}$ is computed at $\mathbf{s}_i = \hat{\mathbf{s}}_{i-1}$.

**Algorithm: Extended Kalman Filter**

- Prediction of states:
  $$\hat{\mathbf{s}}_{i|i-1} \ = \ \mathbf{h}_i(\hat{\mathbf{s}}_{i-1})$$
- Prediction of the covariance matrix of states:
  $$P_{i|i-1} \ = \ \frac{\partial \mathbf{h}_i}{\partial \mathbf{s}_i} P_{i-1} \frac{\partial \mathbf{h}_i}{\partial \mathbf{s}_i}^T \ + \ Q_{i-1}$$
- Kalman gain matrix:
  $$K_i \ = \ P_{i|i-1} M_i^T (M_i P_{i|i-1} M_i^T + \Lambda_{\boldsymbol{\xi}_i})^{-1}$$
- Update of the state estimation:
  $$\begin{aligned}\hat{\mathbf{s}}_i &= \hat{\mathbf{s}}_{i|i-1} + K_i(\mathbf{y}_i - M_i \hat{\mathbf{s}}_{i|i-1}) \\ &= \hat{\mathbf{s}}_{i|i-1} - K_i \mathbf{f}_i(\mathbf{x}_i, \hat{\mathbf{s}}_{i|i-1})\end{aligned}$$
- Update of the covariance matrix of states:
  $$P_i \ = \ (\mathbf{I} - K_i M_i) P_{i|i-1}$$
- Initialization:
  $$P_{0|0} \ = \ \Lambda_{\mathbf{s}_0} \quad \text{and} \quad \hat{\mathbf{s}}_{0|0} \ = \ E[\mathbf{s}_0]$$

## 8.3 Discussion

The above Kalman filter formalism is under the assumptions that the system-noise process and the measurement-noise process are uncorrelated and that they are all Gaussian white noise sequences. These assumptions are adequate in solving the problems addressed in this monograph. In the case that noise processes are correlated or they are not white (i.e., colored), the reader is referred to [3] for the derivation of the Kalman filter equations. The numerical unstability of Kalman filter implementation is well known. Several techniques are developed to overcome those problems, such as square-root filtering and U-D factorization. See [13] for a thorough discussion.

There exist many other methods to solve the parameter estimation problem: general minimization procedures, weighted least-squares method, and the Bayesian decision-theoretic approach. In the appendix to this chapter, we review briefly several least-squares techniques. We choose the Kalman filter approach as our main tool to solve the parameter estimation problem. This is for the following reasons:

- the Kalman filter takes explicitly into account the measurement uncertainties,
- the Kalman filter takes measurements into account incrementally (recursivity),
- the Kalman filter is a simple and efficient procedure to solve the problem (computational tractability),
- the Kalman filter can take into account *a priori* information, if any.

The linearization of a nonlinear model leads to small errors in the estimates, which in general can be neglected, especially if the relative accuracy is better than 10% [15, 4]. However, as pointed by Maybank [12], the extended Kalman filter seriously *underestimates* covariance. Furthermore, if the current estimate $\hat{\mathbf{s}}_{i|i-1}$ is very different from the true one, the first-order approximation, (22 and 23), is not good anymore, and the final estimate given by the filter may be significantly different from the true one. One approach to reduce the effect of nonlinearities is to apply iteratively the Kalman filter (called the *iterated extended Kalman filter*).

## 8.4 Iterated Extended Kalman Filter

The Iterated Extended Kalman Filter (IEKF) could be applied either globally or locally.

The global IEKF is applied to the whole observed data. Given a set of $n$ observations $\{\mathbf{x}_i \ , \ i = 1 \cdots n\}$. The initial state estimate is $\hat{\mathbf{s}}_0$ with covariance matrix

$\Lambda_{\hat{\mathbf{s}}_0}$. After applying the EKF to the set $\{\mathbf{x}_i\}$, we get an estimate $\hat{\mathbf{s}}_n^1$ with covariance matrix $P_n^1$ (the superscript, 1 here, denotes the number of iteration). Before performing the next iteration, we must back propagate $\hat{\mathbf{s}}_n^1$ to time $t_0$, denoted by ${}_0\hat{\mathbf{s}}_n^1$. At iteration 2, ${}_0\hat{\mathbf{s}}_n^1$ is used as the initial state estimate, but the original initial covariance matrix $\Lambda_{\hat{\mathbf{s}}_0}$ is again used as the initial covariance matrix at this iteration. This is because if we use the new covariance matrix, it would mean we have two identical sets of measurements. Due to the requirement of the back propagation of the state estimate, the application of the global IEKF is very limited. Maybe it is interesting only when the state does not evolve over time [1]. In that case, no back propagation is required. In the problem of estimating 3D motion between two frames, the EKF is applied spatially, i.e., it is applied to a number of matches. The 3D motion (the state) does not change from one match to another, thus the global IEKF can be applied.

The local IEKF [7, 14] is applied to a single sample data by redefining the nominal trajectory and relinearizing the measurement equation. It is capable of providing better performance than the basic EKF, especially in the case of significant nonlinearity in the measurement function $\mathbf{f}_i(\mathbf{x'}_i, \mathbf{s})$. This is because when $\hat{\mathbf{s}}_i$ is generated after measurement incorporation, this value can serve as a better state estimate than $\hat{\mathbf{s}}_{i|i-1}$ for evaluating $\mathbf{f}_i$ and $M_i$ in the measurement update relations. Then the state estimate after measurement incorporation could be recomputed, iteratively if desired. Thus, in IEKF, the measurement update relations are replaced by setting $\hat{\mathbf{s}}_i^0 = \hat{\mathbf{s}}_{i|i-1}$ (here, the superscript denotes again the number of iteration) and doing iteration on

$$K_i = P_{i|i-1} M_i^{k\,T} (M_i^k P_{i|i-1} M_i^{k\,T} + \Lambda_{\boldsymbol{\xi}_i})^{-1} , \tag{24}$$

$$\hat{\mathbf{s}}_i^{k+1} = \hat{\mathbf{s}}_i^k - K_i \mathbf{f}_i^k(\mathbf{x}_i, \hat{\mathbf{s}}_i^k) \tag{25}$$

for iteration number $k = 0, 1, \cdots, N-1$ and then setting $\hat{\mathbf{s}}_i = \hat{\mathbf{s}}_i^N$. The iteration could be stopped when consecutive values $\hat{\mathbf{s}}_i^k$ and $\hat{\mathbf{s}}_i^{k+1}$ differ by less than a preselected threshold. The covariance matrix is then updated based on $\hat{\mathbf{s}}_i^N$.

## 8.5 Application to Conic Fitting

Let us choose the normalization with $A + C = 1$ (see Sect.4.1). The state vector can now be defined as:

$$\mathbf{s} = [A, B, D, E, F]^T .$$

The measurement vector is: $\mathbf{x}_i = [x_i, y_i]^T$. As the conic parameters are the same for all points, we have the following simple system equation:

$$\mathbf{s}_{i+1} = \mathbf{s}_i \ ,$$

and the noise term $\mathbf{n}_i$ is zero. The observation function is

$$\mathbf{f}_i(\mathbf{x}_i, \mathbf{s}) = (x_i^2 - y_i^2)A + 2x_iy_iB + 2x_iD + 2y_iE + F + y_i^2 \ .$$

In order to apply the extended Kalman filter, we need to compute the derivative of $\mathbf{f}_i(\mathbf{x}_i, \mathbf{s})$ with respect to $\mathbf{s}$ and that with respect to $\mathbf{x}_i$, which are given by

$$\frac{\partial \mathbf{f}_i(\mathbf{x}_i, \mathbf{s})}{\partial \mathbf{s}} = [x_i^2 - y_i^2,\ 2x_iy_i,\ 2x_i,\ 2y_i\ , 1] \tag{26}$$

$$\frac{\partial \mathbf{f}_i(\mathbf{x}_i, \mathbf{s})}{\partial \mathbf{x}_i} = 2[x_iA + y_iB + D,\ -y_iA + x_iB + E + y_i] \ . \tag{27}$$

# 9 Robust Estimation

## 9.1 Introduction

As have been stated before, least-squares estimators assume that the noise corrupting the data is of zero mean, which yields an *unbiased* parameter estimate. If the noise variance is known, an *minimum-variance* parameter estimate can be obtained by choosing appropriate weights on the data. Furthermore, least-squares estimators implicitly assume that the entire set of data can be interpreted by *only one parameter vector* of a given model. Numerous studies have been conducted, which clearly show that least-squares estimators are vulnerable to the violation of these assumptions. Sometimes even when the data contains only one bad datum, least-squares estimates may be completely perturbed. During the last three decades, many robust techniques have been proposed, which are not very sensitive to departure from the assumptions on which they depend.

Hampel [5] gives some justifications to the use of robustness (quoted in [17]):

> What are the reasons for using robust procedures? There are mainly two observations which combined give an answer. Often in statistics one is using a parametric model implying a very limited set of probability distributions though possible, such as the common model of normally distributed errors, or that of exponentially distributed observations. Classical (parametric) statistics derives results under the assumption that

these models were strictly true. However, apart from some simple discrete models perhaps, such models are never exactly true. We may try to distinguish three main reasons for the derivations: (i) rounding and grouping and other "local inaccuracies"; (ii) the occurrence of "gross errors" such as blunders in measuring, wrong decimal points, errors in copying, inadvertent measurement of a member of a different population, or just "something went wrong"; (iii) the model may have been conceived only as an approximation anyway, e.g. by virtue of the central limit theorem.

If we have some a priori knowledge about the parameters to be estimated, techniques, e.g. Kalman filtering technique, based on the test of *Mahalanobis distance* can be used to yield a robust estimate [24].

In the following, we describe four major approaches to robust estimation.

## 9.2 Clustering or Hough Transform

One of the oldest robust methods used in image analysis and computer vision is the *Hough transform*. The idea is to map data into the parameter space, which is appropriately *quantized*, and then seek for the most likely parameter values to interpret data through *clustering*. A classical example is the detection of straight lines given a set of edge points. In the following, we take the example of estimating plane rigid motion from two sets of points.

Given $p$ 2D points in the first set, noted $\{\mathbf{m}_i\}$, and $q$ 2D points in the second set, noted $\{\mathbf{m}'_j\}$, we must find a rigid transformation between the two sets. The pairing between $\{\mathbf{m}_i\}$ and $\{\mathbf{m}'_j\}$ is assumed not known. A rigid transformation can be uniquely decomposed into a rotation around the origin and a translation, in that order. The corresponding parameter space is three-dimensional: one parameter for the rotation angle $\theta$ and two for the translation vector $\mathbf{t} = [t_x, t_y]^T$. More precisely, if $\mathbf{m}_i$ is paired to $\mathbf{m}'_j$, then

$$\mathbf{m}'_j = \mathbf{R}\mathbf{m}_i + \mathbf{t}$$

with

$$\mathbf{R} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} .$$

It is clear that at least two pairings are necessary for a unique estimate of rigid transformation. The three-dimensional parameter space is quantized as many levels as necessary according to the required precision. The rotation angle $\theta$ ranges from

0 to $2\pi$. We can fix the quantization interval for the rotation angle, say, at $\pi/6$, and we have $N_\theta = 12$ units. The translation is not bounded, but it is in practice. We can assume for example that the translation between two images cannot exceed 200 pixels in each direction (i.e. $t_x, t_y \in [-200, 200]$). If we choose an interval of 20 pixels, then we have $N_{t_x} = N_{t_y} = 20$ units in each direction. The quantized space can then considered as a three-dimensional *accumulator*, of $N = N_\theta N_{t_x} N_{t_y} = 4800$ cells, that is initialized to zero.

Since one pairing of points does not entirely constrain the motion, it is difficult to update the accumulator because the constraint on motion is not simple. Instead, we can match $n$-tuples of points in the first set and in the second set, where $n$ is the smallest value such that matching $n$ points in the first set with $n$ points in the second set completely determines the motion (in our case, $n = 2$). Let $(\mathbf{z}_k, \mathbf{z}'_l)$ be one of such matches, where $\mathbf{z}_k$ and $\mathbf{z}'_l$ are both vectors of dimension $2n$, each composed of $n$ points in the first and second set, respectively. Then the number of matches to be considered is of order of $p^n q^n$ (of course, we do not need to consider all matches in our particular problem, because the distance invariance between a pair of points under rigid transformation can be used to discard the infeasible matches). For each such match, we compute the motion parameters, and the corresponding accumulator cell is increased by 1. After all matches have been considered, peaks in the accumulator indicate the best candidates for the motion parameters.

In general, if the number of data is not larger enough than the number of unknowns, then the maximum peak is not much higher than other peaks, and it may be not the correct one because of data noise and because of parameter space quantization. The Hough transform is thus highly suitable for problems having enough data to support the expected solution.

Because of noise in the measurements, the right peak in the accumulator may be very blurred so that it is not easily detected. The accuracy in the localization with the above simple implementation may be poor. There are several ways to improve it.

- Instead of select just the maximum peak, we can fit a quadratic hyper-surface. The position of its maximum gives a better localization in the parameter space, and the curvature can be used as an indication of the uncertainty of the estimation.

- Statistical clustering techniques can be used to discriminate different candidates of the solution.

- Instead of using an integer accumulator, the uncertainty of data can be taken into account and propagated to the parameter estimation, which would considerably increase the performance.

The Hough transform technique actually follows the principle of *maximum likelihood estimation*. Let $\mathbf{p}$ be the parameter vector ($\mathbf{p} = [\theta, t_x, t_y]^T$ in the above example). Let $\mathbf{x}_m$ be one datum ($\mathbf{x}_m = [\mathbf{z}_k^T, \mathbf{z}_l'^T]^T$ in the above example). Under the assumption that the data $\{\mathbf{x}_m\}$ represent the complete sample of the probability density function of $\mathbf{p}$, $f_{\mathbf{p}}(\mathbf{p})$, we have

$$L(\mathbf{p}) = f_{\mathbf{p}}(\mathbf{p}) = \sum_m f_{\mathbf{p}}(\mathbf{p}|\mathbf{x}_m) \Pr(\mathbf{x}_m)$$

by using the law of total probability. The maximum of $f_{\mathbf{p}}(\mathbf{p})$ is considered as the estimation of $\mathbf{p}$. The Hough transform described above can thus be considered as one approximation.

Because of its nature of global search, the Hough transform technique is rather robust, even when there is a high percentage of gross errors in the data. For better accuracy in the localization of solution, we can increase the number of samples in each dimension of the quantized parameter space. The size of the accumulator increases rapidly with the required accuracy and the number of unknowns. This technique is rarely applied to solve problems having more than three unknowns, and is not suitable for conic fitting.

## 9.3 Regression Diagnostics

Another old robust method is the so-called *regression diagnostics*. It tries to iteratively detect possibly wrong data and reject them through analysis of globally fitted model. The classical approach works as follows:

1. Determine an initial fit to the whole set of data through least squares.
2. Compute the residual for each datum.
3. Reject all data whose residuals exceed a predetermined threshold; if no data have been removed, then stop.
4. Determine a new fit to the remaining data, and goto step 2.

Clearly, the success of this method depends tightly upon the quality of the initial fit. If the initial fit is very poor, then the computed residuals based on it are

meaningless; so is the diagnostics of them for outlier rejection. As pointed out by Barnett and Lewis, with least-squares techniques, *even one or two outliers in a large set can wreak havoc*! This technique thus does not guarantee for a correct solution. However, experiences have shown that this technique works well for problems with a moderate percentage of outliers and more importantly outliers only having *gross errors less than the size of good data.*

The threshold on residuals can be chosen by experiences using for example graphical methods (plotting residuals in different scales). Better is to use a priori statistical noise model of data and a chosen confidence level. Let $r_i$ be the residual of the $i^{\text{th}}$ data, and $\sigma_i$ be the predicted variance of the $i^{\text{th}}$ residual based on the characteristics of the data nose and the fit, the standard test statistics $e_i = r_i/\sigma_i$ can be used. If $e_i$ is not acceptable, the corresponding datum should be rejected.

One improvement to the above technique uses *influence measures* to pinpoint potential outliers. These measures asses the extent to which a particular datum influences the fit by determining the change in the solution when that datum is omitted. The refined technique works as follows:

1. Determine an initial fit to the whole set of data through least squares.
2. Conduct a statistic test whether the measure of fit $f$ (e.g. sum of square residuals) is acceptable; if it is, then stop.
3. For each datum $I$, delete it from the data set and determine the new fit, each giving a measure of fit denoted by $f_i$. Hence determine the change in the measure of fit, i.e. $\Delta f_i = f - f_i$, when datum $i$ is deleted.
4. Delete datum $i$ for which $\Delta f_i$ is the largest, and goto step 2.

It can be shown [22] that the above two techniques agrees with each other at the first order approximation, i.e. the datum with the largest residual is also that datum inducing maximum change in the measure of fit at a first order expansion. The difference is that whereas the first technique simply rejects the datum that deviates most from the current fit, the second technique rejects the point whose exclusion will result in the best fit on the *next* iteration. In other words, the second technique looks ahead to the next fit to see what improvements will actually materialize.

As can be remarked, the regression diagnostics approach depends heavily on a priori knowledge in choosing the thresholds for outlier rejection.

## 9.4 M-estimators

One popular robust technique is the so-called *M-estimators*. Let $r_i$ be the *residual* of the $i^{\text{th}}$ datum, i.e. the difference between the $i^{\text{th}}$ observation and its fitted value. The standard least-squares method tries to minimize $\sum_i r_i^2$, which is unstable if there are outliers present in the data. Outlying data give an effect so strong in the minimization that the parameters thus estimated are distorted. The M-estimators try to reduce the effect of outliers by replacing the squared residuals $r_i^2$ by another function of the residuals, yielding

$$\min \sum_i \rho(r_i) \ , \tag{28}$$

where $\rho$ is a symmetric, positive-definite function with a unique minimum at zero, and is chosen to be less increasing than square. Instead of solving directly this problem, we can implement it as an iterated reweighted least-squares one. Now let us see how.

Let $\mathbf{p} = [p_1, \ldots, p_m]^T$ be the parameter vector to be estimated. The M-estimator of $\mathbf{p}$ based on the function $\rho(r_i)$ is the vector $\mathbf{p}$ which is the solution of the following $m$ equations:

$$\sum_i \psi(r_i) \frac{\partial r_i}{\partial p_j} = 0 \ , \quad \text{for } j = 1, \ldots, m, \tag{29}$$

where the derivative $\psi(x) = \mathrm{d}\rho(x)/\mathrm{d}x$ is called the *influence function*. If now we define a *weight function*

$$w(x) = \frac{\psi(x)}{x} \ , \tag{30}$$

then Equation (29) becomes

$$\sum_i w(r_i) r_i \frac{\partial r_i}{\partial p_j} = 0 \ , \quad \text{for } j = 1, \ldots, m. \tag{31}$$

This is exactly the system of equations that we obtain if we solve the following iterated reweighted least-squares problem

$$\min \sum_i w(r_i^{(k-1)}) r_i^2 \ , \tag{32}$$

where the superscript $^{(k)}$ indicates the iteration number. The weight $w(r_i^{(k-1)})$ should be recomputed after each iteration in order to be used in the next iteration.

The influence function $\psi(x)$ measures the influence of a datum on the value of the parameter estimate. For example, for the least-squares with $\rho(x) = x^2/2$, the influence function is $\psi(x) = x$, that is, the influence of a datum on the estimate increases linearly with the size of its error, which confirms the non-robusteness of the least-squares estimate. When an estimator is robust, it may be inferred that the influence of any single observation (datum) is insufficient to yield any significant offset [17]. There are several constraints that a robust $M$-estimator should meet:

- The first is of course to have a bounded influence function.
- The second is naturally the requirement of the robust estimator to be unique. This implies that the objective function of parameter vector $\mathbf{p}$ to be minimized should have a unique minimum. This requires that *the individual $\rho$-function is* convex *in variable* $\mathbf{p}$. This is necessary because only requiring a $\rho$-function to have a unique minimum is not sufficient. This is the case with maxima when considering mixture distribution; the sum of unimodal probability distributions is very often multimodal. The convexity constraint is equivalent to imposing that $\frac{\partial^2 \rho(.)}{\partial \mathbf{p}^2}$ is non-negative definite.
- The third one is a practical requirement. Whenever $\frac{\partial^2 \rho(.)}{\partial \mathbf{p}^2}$ is singular, the objective should have a gradient, i.e. $\frac{\partial \rho(.)}{\partial \mathbf{p}} \neq \mathbf{0}$. This avoids having to search through the complete parameter space.

Table 1 lists a few commonly used influence functions. They are graphically dipicted in Fig. 4. Note that not all these functions satisfy the above requirements.

Briefly we give a few indications of these functions:

- $L_2$ (i.e. least-squres) estimators are not robust because their influence function is not bounded.
- $L_1$ (i.e. absolute value) estimators are not stable because the $\rho$-function $|x|$ is not trictly convex in $x$. Indeed, the second derivative at $x = 0$ is unbounded, and an indeterminant solution may result.
- $L_1$ estimators reduce the influence of large errors, but they still have an influence because the influence function has no cut off point.

Table 1: A few commonly used M-estimators

| type | $\rho(x)$ | $\psi(x)$ | $w(x)$ |
|---|---|---|---|
| $L_2$ | $x^2/2$ | $x$ | $1$ |
| $L_1$ | $\lvert x\rvert$ | $\operatorname{sgn}(x)$ | $\dfrac{1}{\lvert x\rvert}$ |
| $L_1-L_2$ | $2(\sqrt{1+x^2/2}-1)$ | $\dfrac{x}{\sqrt{1+x^2/2}}$ | $\dfrac{1}{\sqrt{1+x^2/2}}$ |
| $L_p$ | $\dfrac{\lvert x\rvert^\nu}{\nu}$ | $\operatorname{sgn}(x)\lvert x\rvert^{\nu-1}$ | $\lvert x\rvert^{\nu-2}$ |
| "Fair" | $c^2[\dfrac{\lvert x\rvert}{c}-\log(1+\dfrac{\lvert x\rvert}{c})]$ | $\dfrac{x}{1+\lvert x\rvert/c}$ | $\dfrac{1}{1+\lvert x\rvert/c}$ |
| Huber $\begin{cases}\text{if } \lvert x\rvert\le k\\ \text{if } \lvert x\rvert\ge k\end{cases}$ | $\begin{cases}x^2/2\\ k(\lvert x\rvert-k/2)\end{cases}$ | $\begin{cases}x\\ k\operatorname{sgn}(x)\end{cases}$ | $\begin{cases}1\\ k/\lvert x\rvert\end{cases}$ |
| Cauchy | $\dfrac{c^2}{2}\log(1+(x/c)^2)$ | $\dfrac{x}{1+(x/c)^2}$ | $\dfrac{1}{1+(x/c)^2}$ |
| Geman-McClure | $\dfrac{x^2/2}{1+x^2}$ | $\dfrac{x}{(1+x^2)^2}$ | $\dfrac{1}{(1+x^2)^2}$ |
| Welsch | $\dfrac{c^2}{2}[1-\exp(-(x/c)^2)]$ | $x\exp(-(x/c)^2)$ | $\exp(-(x/c)^2))$ |
| Tukey $\begin{cases}\text{if } \lvert x\rvert\le c\\ \text{if } \lvert x\rvert> c\end{cases}$ | $\begin{cases}\dfrac{c^2}{6}\left(1-[1-(x/c)^2]^3\right)\\ (c^2/6)\end{cases}$ | $\begin{cases}x[1-(x/c)^2]^2\\ 0\end{cases}$ | $\begin{cases}[1-(x/c)^2]^2\\ 0\end{cases}$ |

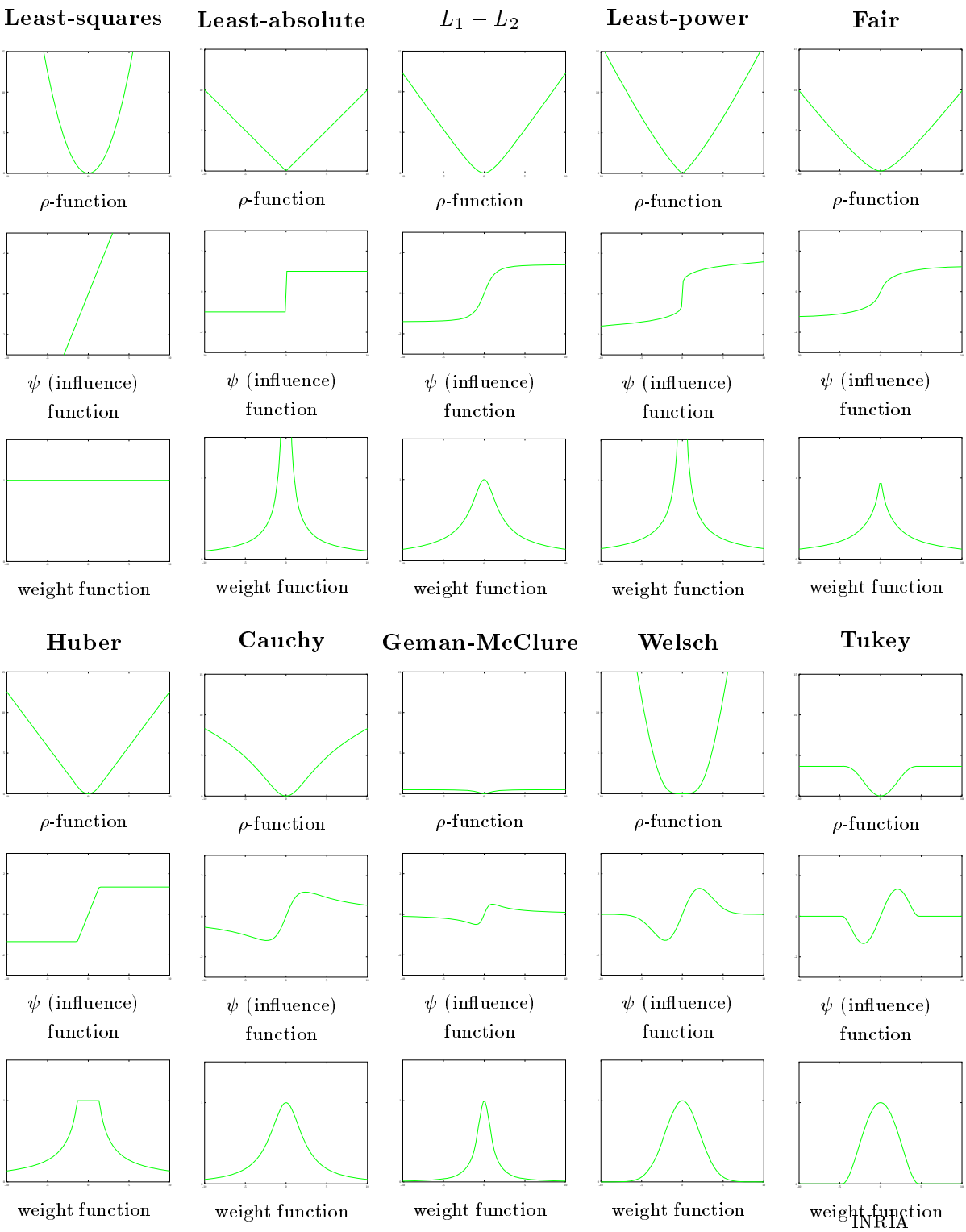


Fig. 4: Graphic representations of a few common M-estimators

- $L_1 - L_2$ estimators take both the advantage of the $L_1$ estimators to reduce the influence of large errors and that of $L_2$ estimators to be convex.

- The $L_p$ (*least-powers*) function represents a family of functions. It is $L_2$ with $\nu = 2$ and $L_1$ with $\nu = 1$. The smaller $\nu$, the smaller is the incidence of large errors in the estimate $\mathbf{p}$. It appears that $\nu$ must be fairly moderate to provide a relatively robust estimator or, in other words, to provide an estimator scarcely perturbed by outlying data. The selection of an optimal $\nu$ has been investigated, and for $\nu$ around 1.2, a good estimate may be expected [17]. However, many difficulties are encountered in the computation when parameter $\nu$ is in the range of interest $1 < \nu < 2$, because zero residuals are troublesome.

- The function "Fair" is among the possibilities offered by the Roepack package (see [17]). It has everywhere defined continuous derivatives of first three orders, and yields a unique solution. The 95% asymptotic efficiency on the standard normal distribution is obtained with the tuning constan $c = 1.3998$.

- Huber's function [6] is a parabola in the vicinity of zero, and increases linearly at a given level $|x| > k$. The 95% asymptotic efficiency on the standard normal distribution is obtained with the tuning constant $k = 1.345$. This estimator is so satisfactory that it has been recommended for almost all situations; very rarely it has been found to be inferior to some other $\rho$-function. However, from time to time, difficulties are encountered, which may be due to the lack of stability in the gradient values of the $\rho$-function because of its *discontinuous second derivative*:

$$\frac{\mathrm{d}^2\rho(x)}{\mathrm{d}x^2} = \begin{cases} 1 & \text{if } |x| \leq k, \\ 0 & \text{if } |x| \geq k. \end{cases}$$

  The modification proposed in [17] is the following

$$\rho(x) = \begin{cases} c^2[1 - \cos(x/c)] & \text{if } |x|/c \leq \pi/2, \\ c|x| + c^2(1 - \pi/2) & \text{if } |x|/c \geq \pi/2. \end{cases}$$

  The 95% asymptotic efficiency on the standard normal distribution is obtained with the tuning constan $c = 1.2107$.

- Cauchy's function, also known as the Lorentzian function, does not guarantee a unique solution. With a descending first derivative, such a function has a tendency to yield erroneous solutions in a way which cannot be observed. The

95% asymptotic efficiency on the standard normal distribution is obtained with the tuning constan $c = 2.3849$.

- The other remaining functions have the same problem as the Cauchy function. As can be seen from the influence function, the influence of large errors only decreases linearily with their size. The Geman-McClure and Welsh functions try to further reduce the effect of large errors, and the Tukey's biweight function even suppress the outliers. The 95% asymptotic efficiency on the standard normal distribution of the Tukey's biweight function is obtained with the tuning constant $c = 4.6851$; that of the Welsch function, with $c = 2.9846$.

There still exist many other $\rho$-functions, such as Andrew's cosine wave function. Another commonly used function is the following tri-weight one:

$$w_i = \begin{cases} 1 & |r_i| \leq \sigma \\ \sigma/|r_i| & \sigma < |r_i| \leq 3\sigma \\ 0 & 3\sigma < |r_i| \end{cases},$$

where $\sigma$ is some estimated standard deviation of errors.

It seems difficult to select a $\rho$-function for general use without being rather arbitrary. Following Rey [17], for the location (or regression) problems, the best choice is the $L_p$ in spite of its theoretical non-robustness: they are quasi-robust. However, it suffers from its computational difficulties. The second best function is "Fair", which can yield nicely converging computational procedures. Eventually comes the Huber's function (either original or modified form). All these functions do not eliminate completely the influence of large gross errors.

The four last functions do not quarantee unicity, but reduce considerably, or even eliminate completely, the influence of large gross errors. As proposed by Huber [6], one can start the iteration process with a convex $\rho$-function, iterate until convergence, and then apply a few iterations with one of those non-convex functions to eliminate the effect of large errors.

## 9.5 Least Median of Squares

The least-median-of-squares (LMedS) method estimates the parameters by solving the nonlinear minimization problem:

$$\min \underset{i}{\text{med}}\, r_i^2 .$$

That is, the estimator must yield the smallest value for the median of squared residuals computed for the entire data set. It turns out that this method is very robust to false matches as well as outliers due to bad localization. Unlike the M-estimators, however, the LMedS problem cannot be reduced to a weighted least-squares problem. It is probably impossible to write down a straightforward formula for the LMedS estimator. It must be solved by a search in the space of possible estimates generated from the data. Since this space is too large, only a randomly chosen subset of data can be analyzed. The algorithm which we describe below for robustly estimating a conic follows that structured in [21, Chap. 5], as outlined below.

Given $n$ points: $\{\mathbf{m}_i = [x_i, y_i]^T\}$.

1. A Monte Carlo type technique is used to draw $m$ random subsamples of $p$ different points. For the problem at hand, we select *five* (i.e. $p = 5$) points because we need at least five points to define a conic.

2. For each subsample, indexed by $J$, we use any of the techniques described in Sect.4 to compute the conic parameters $\mathbf{p}_J$. (Which technique is used is not important because an exact solution is possible for five different points.)

3. For each $\mathbf{p}_J$, we can determine the median of the squared residuals, denoted by $M_J$, with respect to the whole set of points, i.e.

$$M_J = \underset{i=1,\ldots,n}{\text{med}} r_i^2(\mathbf{p}_J, \mathbf{m}_i) \ .$$

   Here, we have a number of choices for $r_i(\mathbf{p}_J, \mathbf{m}_i)$, the residual of the $i^{\text{th}}$ point with respect to the conic $\mathbf{p}_J$. Depending on the demanding precision, computation requirement, etc., one can use the algebraic distance, the Euclidean distance, or the gradient weighted distance.

4. We retain the estimate $\mathbf{p}_J$ for which $M_J$ is minimal among all $m$ $M_J$'s.

The question now is: *How do we determine* $m$*?* A subsample is "good" if it consists of $p$ good data points. Assuming that the whole set of points may contain up to a fraction $\varepsilon$ of outliers, the probability that at least one of the $m$ subsamples is good is given by

$$P = 1 - [1 - (1 - \varepsilon)^p]^m \ . \tag{33}$$

By requiring that $P$ must be near 1, one can determine $m$ for given values of $p$ and $\varepsilon$:

$$m = \frac{\log(1-P)}{\log[1-(1-\varepsilon)^p]} .$$

In our implementation, we assume $\varepsilon = 40\%$ and require $P = 0.99$, thus $m = 57$. Note that the algorithm can be speeded up considerably by means of parallel computing, because the processing for each subsample can be done independently.

As noted in [21], the LMedS *efficiency* is poor in the presence of Gaussian noise. The efficiency of a method is defined as the ratio between the lowest achievable variance for the estimated parameters and the actual variance provided by the given method. To compensate for this deficiency, we further carry out a weighted least-squares procedure. The *robust standard deviation* estimate is given by

$$\hat{\sigma} = 1.4826[1 + 5/(n-p)]\sqrt{M_J} ,$$

where $M_J$ is the minimal median. The constant 1.4826 is a coefficient to achieve the same efficiency as a least-squares in the presence of only Gaussian noise; $5/(n-p)$ is to compensate the effect of a small set of data. The reader is referred to [21, page 202] for the details of these magic numbers. Based on $\hat{\sigma}$, we can assign a weight for each correspondence:

$$w_i = \begin{cases} 1 & \text{if } r_i^2 \leq (2.5\hat{\sigma})^2 \\ 0 & \text{otherwise} , \end{cases}$$

where $r_i$ is the residual of the $i^{\text{th}}$ point with respect to the conic $\mathbf{p}$. The correspondences having $w_i = 0$ are outliers and should not be further taken into account. The conic $\mathbf{p}$ is finally estimated by solving the weighted least-squares problem:

$$\min_{\mathbf{p}} \sum_i w_i r_i^2$$

using one of the numerous techniques described before. We have thus robustly estimated the conic because outliers have been detected and discarded by the LMedS method.

As said previously, computational efficiency of the LMedS method can be achieved by applying a Monte-Carlo type technique. However, the five points of a subsample thus generated may be very close to each other. Such a situation should be avoided because the estimation of the conic from such points is highly instable and

the result is useless. It is a waste of time to evaluate such a subsample. In order to achieve higher stability and efficiency, we develop a *regularly random selection method* based on bucketing techniques, which works as follows. We first calculate the `min` and `max` of the coordinates of the points in the first image. The region is then evenly divided into $b \times b$ buckets (see Fig. 5; in our implementation, $b = 8$). To each bucket is attached a set of points, and indirectly a set of matches, which fall in it. The buckets having no matches attached are excluded. To generate a subsample of 5 points, we first randomly select 5 mutually different buckets, and then randomly choose one match in each selected bucket.

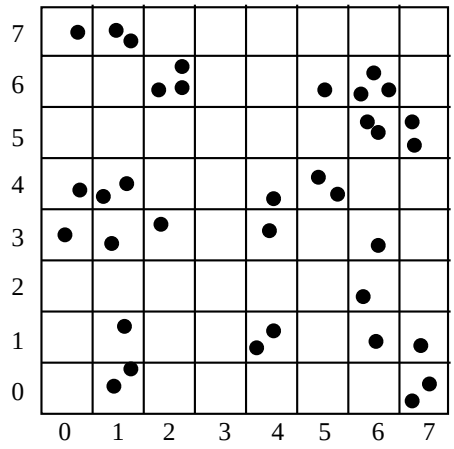


Fig. 5: Illustration of a bucketing technique

One question remains: How many subsamples are required? If we assume that bad points are uniformly distributed in space, and if each bucket has the same number of points and the random selection is uniform, the formula (33) still holds. However, the number of points in one bucket may be quite different from that in another. As a result, a point belonging to a bucket having fewer points has a higher probability to be selected. It is thus preferred that a bucket having many points has a higher probability to be selected than a bucket having few points, in order that each point has almost the same probability to be selected. This can be realized by the following procedure. If we have in total $l$ buckets, we divide [0 1] into $l$ intervals such that the width of the $i^{\text{th}}$ interval is equal to $n_i / \sum_i n_i$, where $n_i$ is the number of points attached to the $i^{\text{th}}$ bucket (see Fig. 6). During the bucket selection procedure, a number, produced by a [0 1] uniform random generator, falling in the $i^{\text{th}}$ interval implies that the $i^{\text{th}}$ bucket is selected.

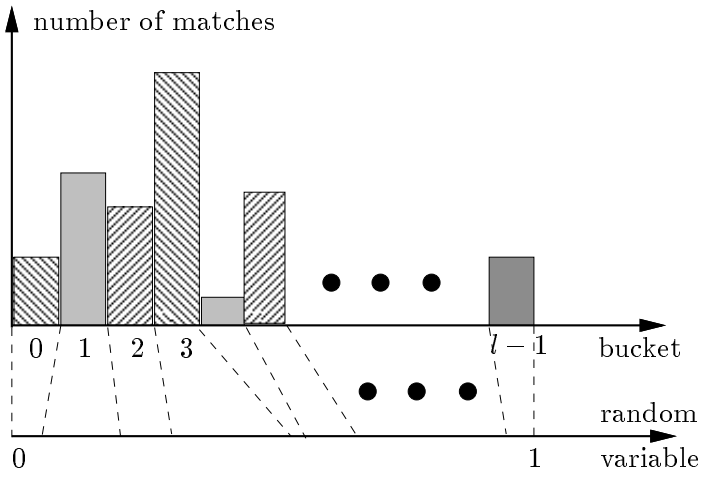


Fig. 6: Interval and bucket mapping

We have applied this technique to matching between two uncalibrated images [23]. Given two uncalibrated images, the only available geometric constraint is the *epipolar constraint*. The idea underlying our approach is to use classical techniques (correlation and relaxation methods in our particular implementation) to find an initial set of matches, and then use the Least Median of Squares (LMedS) to discard false matches in this set. The epipolar geometry can then be accurately estimated using a meaningful image criterion. More matches are eventually found, as in stereo matching, by using the recovered epipolar geometry.

## 10 Conclusions

In this tutorial, I have presented what is probably the most commonly used techniques for parameter estimation in computer vision. Particular attention has been devoted to discussions about the choice of appropriate minimization criteria and the robustness of the different techniques. Hopefully, the reader will find this tutorial useful. Comments are extremely welcome.

Another technique, which I consider to be very important and becomes popular now in Computer Vision, is the *Minimum Description Length* (MDL) principle. However, since I have not yet myself applied it to solve any problem, I am not in a position to present it. The reader is referred to [18, 9, 10].

## References

[1] N. Ayache. *Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception.* MIT Press, Cambridge, MA, 1991.

[2] J.V. Beck and K.J. Arnold. *Parameter estimation in engineering and science.* Wiley series in probability and mathematical statistics. J. Wiley, New York, 1977.

[3] C.K. Chui and G. Chen. *Kalman Filtering with Real-Time Applications.* Springer Ser. Info. Sci., Vol. 17. Springer, Berlin, Heidelberg, 1987.

[4] W. Förstner. Reliability analysis of parameter estimation in linear models with application to mensuration problems in computer vision. *Comput. Vision, Graphics Image Process.*, 40:273–310, 1987.

[5] F.R. Hampel. Robust estimation: A condensed partial survey. *Z. Wahrscheinlichkeitstheorie Verw. Gebiete*, 27:87–104, 1973.

[6] P.J. Huber. *Robust Statistics.* John Wiley & Sons, New York, 1981.

[7] A.M. Jazwinsky. *Stochastic Processes and Filtering Theory.* Academic, New York, 1970.

[8] K. Kanatani. Renormalization for unbiased estimation. In *Proc. Fourth Int'l Conf. Comput. Vision*, pages 599–606, Berlin, 1993.

[9] Y.G. Leclerc. Constructing simple stable description for image partitioning. *The International Journal of Computer Vision*, 3(1):73–102, 1989.

[10] M. Li. Minimum description length based 2D shape description. In *Proceedings of the 4th Proc. International Conference on Computer Vision*, pages 512–517, Berlin, Germany, May 1993. IEEE Computer Society Press.

[11] D. G. Lowe. Review of "TINA: The Sheffield AIVRU vision system" by J. Porrill et al. In O. Khatib, J. Craig, and T. Lozano-Pérez, editors, *The Robotics Review I*, pages 195–198. MIT Press, Cambridge, MA, 1989.

[12] S.J. Maybank. Filter based estimates of depth. In *Proc. British Machine Vision Conf.*, pages 349–354, University of Oxford, London, UK, September 1990.

[13] P.S. Maybeck. *Stochastic Models, Estimation and Control*, volume 1. Academic, New York, 1979.

[14] P.S. Maybeck. *Stochastic Models, Estimation and Control*, volume 2. Academic, New York, 1982.

[15] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, 1965.

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

[17] William J.J. Rey. *Introduction to Robust and Quasi-Robust Statistical Methods*. Springer, Berlin, Heidelberg, 1983.

[18] J. Rissanen. Minimum description length principle. *Encyclopedia of Statistic Sciences*, 5:523–527, 1987.

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

[20] P.L. Rosin and G.A.W. West. Segmenting curves into elliptic arcs and straight lines. In *Proc. Third Int'l Conf. Comput. Vision*, pages 75–78, Osaka, Japan, 1990.

[21] P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, New York, 1987.

[22] L.S. Shapiro. *Affine Analysis of Image Sequences*. PhD thesis, Dept. of Engineering Science, Oxford University, 1993.

[23] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal*, 1995. to appear. Also INRIA Research Report No.2273, May 1994.

[24] Z. Zhang and O. Faugeras. *3D Dynamic Scene Analysis: A Stereo Based Approach*. Springer, Berlin, Heidelberg, 1992.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399