ROBUST TECHNIQUES

FOR COMPUTER VISION

Peter Meer
Electrical and Computer Engineering Department
Rutgers University

This is a chapter from the upcoming book *Emerging Topics in Computer Vision*, Gerard Medioni and Sing Bing Kang (Eds.), Prentice Hall, 2004.

CONTENTS

Chapter 4

ROBUST TECHNIQUES FOR COMPUTER VISION

4.1 Robustness in Visual Tasks

Visual information makes up about seventy five percent of all the sensorial information received by a person during a lifetime. This information is processed not only efficiently but also transparently. Our awe of visual perception was perhaps the best captured by the seventeenth century british essayist Joseph Addison in an essay on imagination [1].

> Our sight is the most perfect and most delightful of all our senses. It fills the mind with the largest variety of ideas, converses with its objects at the greatest distance, and continues the longest in action without being tired or satiated with its proper enjoyments.

The ultimate goal of computer vision is to mimic human visual perception. Therefore, in the broadest sense, robustness of a computer vision algorithm is judged against the performance of a human observer performing an equivalent task. In this context, robustness is the ability to extract the visual information of relevance for a specific task, even when this information is carried only by a small subset of the data, and/or is significantly different from an already stored representation.

To understand why the performance of generic computer vision algorithms is still far away from that of human visual perception, we should consider the hierarchy of computer vision tasks. They can be roughly classified into three large categories:

- *low level*, dealing with extraction from a single image of salient simple features, such as edges, corners, homogeneous regions, curve fragments;
- *intermediate level*, dealing with extraction of semantically relevant characteristics from one or more images, such as grouped features, depth, motion information;
- *high level*, dealing with the interpretation of the extracted information.

A similar hierarchy is difficult to distinguish in human visual perception, which appears as a single integrated unit. In the visual tasks performed by a human observer an extensive top-down information flow carrying representations derived at higher levels seems to control the processing at lower levels. See [85] for a discussion on the nature of these interactions.

A large amount of psychophysical evidence supports this "closed loop" model of human visual perception. Preattentive vision phenomena, in which salient information pops-out from the image, e.g., [55], [110], or perceptual constancies, in which changes in the appearance of a familiar object are attributed to external causes [36, Chap.9], are only some of the examples. Similar behavior is yet to be achieved in generic computer vision techniques. For example, preattentive vision type processing seems to imply that a region of interest is delineated *before* extracting its salient features.

To approach the issue of robustness in computer vision, will start by mentioning one of the simplest perceptual constancies, the shape constancy. Consider a door opening in front of an observer. As the door opens, its image changes from a rectangle to a trapezoid but the observer will report only the movement. That is, *additional information* not available in the input data was also taken into account. We *know* that a door is a rigid structure, and therefore it is very unlikely that its image changed due to a nonrigid transformation. Since the perceptual constancies are based on rules embedded in the visual system, they can be also deceived. A well known example is the Ames room in which the rules used for perspective foreshortening compensation are violated [36, p.241].

The previous example does not seem to reveal much. Any computer vision algorithm of rigid motion recovery is based on a similar approach. However, the example emphasizes that the employed rigid motion model is only associated with the data and is not intrinsic to it. We could use a completely different model, say of nonrigid doors, but the result would not be satisfactory. Robustness thus is closely related to the availability of a model adequate for the goal of the task.

In today's computer vision algorithms the information flow is almost exclusively bottom-up. Feature extraction is followed by grouping into semantical primitives, which in turn is followed by a task specific interpretation of the ensemble of primitives. The lack of top-down information flow is arguably the main reason why computer vision techniques cannot yet autonomously handle visual data under a wide range of operating conditions. This fact is well understood in the vision community and different approaches were proposed to simulate the top-down information stream.

The increasingly popular Bayesian paradigm is such an attempt. By using a probabilistic representation for the possible outcomes, multiple hypotheses are incorporated into the processing, which in turn guide the information recovery. The dependence of the procedure on the accuracy of the employed representation is relaxed in the semiparametric or nonparametric Bayesian methods, such as particle filtering for motion problems [51]. Incorporating a learning component into computer vision techniques, e.g., [3], [29], is another, somewhat similar approach to use higher level information during the processing.

Comparison with human visual perception is not a practical way to arrive to a definition of robustness for computer vision algorithms. For example, robustness in the context of the human visual system extends to abstract concepts. We can recognize a chair independent of its design, size or the period in which it was made. However, in a somewhat similar experiment, when an object recognition system was programmed to decide if a simple drawing represents a chair, the results were rather mixed [97].

We will not consider high level processes when examining the robustness of vision algorithms, neither will discuss the role of top-down information flow. A computer vision

algorithm will be called robust if it can tolerate outliers, i.e., data which does not obey the assumed model. This definition is similar to the one used in statistics for robustness [40, p.6]

> In a broad informal sense, robust statistics is a body of knowledge, partly formalized into "theories of statistics," relating to deviations from idealized assumptions in statistics.

Robust techniques are used in computer vision for at least thirty years. In fact, those most popular today are related to old methods proposed to solve specific image understanding or pattern recognition problems. Some of them were rediscovered only in the last few years.

The best known example is the Hough transform, a technique to extract multiple instances of a low-dimensional manifold from a noisy background. The Hough transform is a US Patent granted in 1962 [47] for the detection of linear trajectories of subatomic particles in a bubble chamber. In the rare cases when Hough transform is explicitly referenced this patent is used, though an earlier publication also exists [46]. Similarly, the most popular robust regression methods today in computer vision belong to the family of random sample consensus (RANSAC), proposed in 1980 to solve the perspective n-point problem [25]. The usually employed reference is [26]. An old pattern recognition technique for density gradient estimation proposed in 1975 [32], the mean shift, recently became a widely used methods for feature space analysis. See also [31, p.535].

In theoretical statistics, investigation of robustness started in the early 1960s, and the first robust estimator, the M-estimator, was introduced by Huber in 1964. See [49] for the relevant references. Another popular family of robust estimators, including the least median of squares (LMedS), was introduced by Rousseeuw in 1984 [87]. By the end of 1980s these robust techniques became known in the computer vision community.

Application of robust methods to vision problems was restricted at the beginning to replacing a nonrobust parameter estimation module with its robust counterpart, e.g., [4], [41], [59], [103]. See also the review paper [75]. While this approach was successful in most of the cases, soon also some failures were reported [78]. Today we know that these failures are due to the inability of most robust estimators to handle data in which more than one structure is present [9], [98], a situation frequently met in computer vision but almost never in statistics. For example, a window operator often covers an image patch which contains two homogeneous regions of almost equal sizes, or there can be several independently moving objects in a visual scene.

Large part of today's robust computer vision toolbox is indigenous. There are good reasons for this. The techniques imported from statistics were designed for data with characteristics significantly different from that of the data in computer vision. If the data does not obey the assumptions implied by the method of analysis, the desired performance may not be achieved. The development of robust techniques in the vision community (such as RANSAC) were motivated by applications. In these techniques the user has more freedom to adjust the procedure to the specific data than in a similar technique taken from the statistical literature (such as LMedS). Thus, some of the theoretical limitations of a robust method can be alleviated by data specific tuning, which sometimes resulted in attributing

better performance to a technique than is theoretically possible in the general case.

A decade ago, when a vision task was solved with a robust technique, the focus of the research was on the methodology and not on the application. Today the emphasis has changed, and often the employed robust techniques are barely mentioned. It is no longer of interest to have an exhaustive survey of "robust computer vision". For some representative results see the review paper [99] or the special issue [95].

The goal of this chapter is to focus on the theoretical foundations of the robust methods in the context of computer vision applications. We will provide a unified treatment for most estimation problems, and put the emphasis on the underlying concepts and not on the details of implementation of a specific technique. Will describe the assumptions embedded in the different classes of robust methods, and clarify some misconceptions often arising in the vision literature. Based on this theoretical analysis new robust methods, better suited for the complexity of computer vision tasks, can be designed.

4.2 Models and Estimation Problems

In this section we examine the basic concepts involved in parameter estimation. We describe the different components of a model and show how to find the adequate model for a given computer vision problem. Estimation is analyzed as a generic problem, and the differences between nonrobust and robust methods are emphasized. We also discuss the role of the optimization criterion in solving an estimation problem.

4.2.1 Elements of a Model

The goal of data analysis is to provide for data spanning a very high-dimensional space an equivalent low-dimensional representation. A set of measurements consisting of n data vectors $\mathbf{y}_i \in \mathcal{R}^q$ can be regarded as a point in $\mathcal{R}^{nq}$. If the data can be described by a model with only $p \ll nq$ parameters, we have a much more compact representation. Should new data points become available, their relation to the initial data then can be established using only the model. A model has two main components

- the constraint equation;
- the measurement equation.

The constraint describes our a priori knowledge about the nature of the process generating the data, while the measurement equation describes the way the data was obtained.

In the general case a constraint has two levels. The first level is that of the quantities providing the input into the estimation. These *variables* $\{y_1, \ldots, y_q\}$ can be obtained either by direct measurement or can be the output of another process. The variables are grouped together in the context of the process to be modeled. Each ensemble of values for the q variables provides a single input data point, a q-dimensional vector $\mathbf{y} \in \mathcal{R}^q$.

At the second level of a constraint the variables are combined into *carriers*, also called as *basis functions*

$$x_j = \varphi_j(y_1, \ldots, y_q) = \varphi_j(\mathbf{y}) \qquad j = 1, \ldots, m \; . \tag{4.2.1}$$

A carrier is usually a simple nonlinear function in a subset of the variables. In computer vision most carriers are monomials.

The *constraint* is a set of algebraic expressions in the carriers and the parameters $\theta_0, \theta_1, \ldots, \theta_p$

$$\psi_k(x_1, \ldots, x_m; \theta_0, \theta_1, \ldots, \theta_p) = 0 \qquad k = 1, \ldots, K \; . \tag{4.2.2}$$

One of the goals of the estimation process is to find the values of these parameters, i.e., to mold the constraint to the available measurements.

The constraint captures our a priori knowledge about the physical and/or geometrical relations underlying the process in which the data was generated, Thus, the constraint is valid only for the true (uncorrupted) values of the variables. In general these values are not available. The estimation process replaces in the constraint the true values of the variables with their *corrected* values, and the true values of the parameters with their estimates. We will return to this issue in Section 4.2.2.

The expression of the constraint (4.2.2) is too general for our discussion and we will only use a scalar (univariate) constraint, i.e., $K = 1$, which is *linear in the carriers and the parameters*

$$\alpha + \mathbf{x}^\top \boldsymbol{\theta} = 0 \qquad \mathbf{x}^\top = [\varphi_1(\mathbf{y}) \;\; \cdots \;\; \varphi_p(\mathbf{y})] \tag{4.2.3}$$

where the parameter θ_0 associated with the constant carrier was renamed α, all the other carriers were gathered into the vector $\mathbf{x}$, and the parameters into the vector $\boldsymbol{\theta}$. The linear structure of this model implies that $m = p$. Note that the constraint (4.2.3) in general is *nonlinear in the variables*.

The parameters α and $\boldsymbol{\theta}$ are defined in (4.2.3) only up to a multiplicative constant. This ambiguity can be eliminated in many different ways. We will show in Section 4.2.6 that often it is advantageous to impose $\|\boldsymbol{\theta}\| = 1$. Any condition additional to (4.2.3) is called an *ancillary constraint*.

In some applications one of the variables has to be singled out. This variable, denoted z, is called the *dependent variable* while all the other ones are *independent variables* which enter into the constraint through the carriers. The constraint becomes

$$z = \alpha + \mathbf{x}^\top \boldsymbol{\theta} \tag{4.2.4}$$

and the parameters are no longer ambiguous.

To illustrate the role of the variables and carriers in a constraint, will consider the case of the ellipse (Figure 4.1). The constraint can be written as

$$(\mathbf{y} - \mathbf{y}_c)^\top \mathrm{Q} (\mathbf{y} - \mathbf{y}_c) - 1 = 0 \tag{4.2.5}$$

where the two variables are the coordinates of a point on the ellipse, $\mathbf{y}^\top = [y_1 \; y_2]$. The constraint has five parameters. The two coordinates of the ellipse center $\mathbf{y}_c$, and the three distinct elements of the 2×2 symmetric, positive definite matrix Q. The constraint is rewritten under the form (4.2.3) as

$$\alpha + \theta_1 y_1 + \theta_2 y_2 + \theta_3 y_1^2 + \theta_4 y_1 y_2 + \theta_5 y_2^2 = 0 \tag{4.2.6}$$

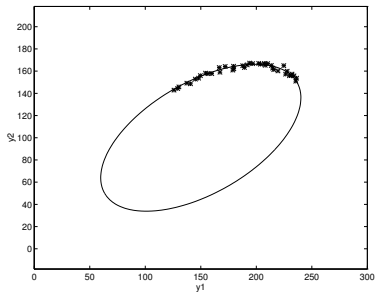


Figure 4.1. A typical nonlinear regression problem. Estimate the parameters of the ellipse from the noisy data points.

where

$$\alpha = \mathbf{y}_c^\top \mathrm{Q} \mathbf{y}_c - 1 \qquad \boldsymbol{\theta}^\top = [-2\mathbf{y}_c^\top \mathrm{Q} \quad Q_{11} \quad 2Q_{12} \quad Q_{22}] \; . \tag{4.2.7}$$

Three of the five carriers

$$\mathbf{x}^\top = [y_1 \quad y_2 \quad y_1^2 \quad y_1 y_2 \quad y_2^2] \tag{4.2.8}$$

are nonlinear functions in the variables.

Ellipse estimation uses the constraint (4.2.6). This constraint, however, has not five but six parameters which are again defined only up to a multiplicative constant. Furthermore, the same constraint can also represent two other conics: a parabola or a hyperbola. The ambiguity of the parameters therefore is eliminated by using the ancillary constraint which enforces that the quadratic expression (4.2.6) represents an ellipse

$$4\theta_3\theta_5 - \theta_4^2 = 1 \; . \tag{4.2.9}$$

The nonlinearity of the constraint in the variables makes ellipse estimation a difficult problem. See [27], [57], [72], [117] for different approaches and discussions.

For most of the variables only the noise corrupted version of their true value is available. Depending on the nature of the data, the noise is due to the measurement errors, or to the inherent uncertainty at the output of another estimation process. While for convenience we will use the term measurements for any input into an estimation process, the above distinction about the origin of the data should be kept in mind.

The general assumption in computer vision problems is that the noise is additive. Thus, the *measurement equation* is

$$\mathbf{y}_i = \mathbf{y}_{io} + \delta\mathbf{y}_i \qquad \mathbf{y}_i \in \mathcal{R}^q \qquad i = 1, \ldots, n \tag{4.2.10}$$

where $\mathbf{y}_{io}$ is the true value of $\mathbf{y}_i$, the i-th measurement. The subscript 'o' denotes the true value of a measurement. Since the constraints (4.2.3) or (4.2.4) capture our a priori knowledge, they are valid for the true values of the measurements or parameters, and should have been written as

$$\alpha + \mathbf{x}_o^\top \boldsymbol{\theta} = 0 \qquad \text{or} \qquad z_o = \alpha + \mathbf{x}_o^\top \boldsymbol{\theta} \tag{4.2.11}$$

where $\mathbf{x}_o = \mathbf{x}(\mathbf{y}_o)$. In the ellipse example y_{1o} and y_{2o} should have been used in (4.2.6).

The noise corrupting the measurements is assumed to be independent and identically distributed (i.i.d.)

$$\delta\mathbf{y}_i \sim GI(\mathbf{0}, \sigma^2 \mathrm{C}_y) \tag{4.2.12}$$

where $GI(\cdot)$ stands for a general symmetric distribution of independent outcomes. Note that this distribution does not necessarily has to be normal. A warning is in order, though. By characterizing the noise only with its first two central moments we implicitly agree to normality, since only the normal distribution is defined uniquely by these two moments.

The independency assumption usually holds when the input data points are physical measurements, but may be violated when the data is the output of another estimation process. It is possible to take into account the correlation between two data points $\mathbf{y}_i$ and $\mathbf{y}_j$ in the estimation, e.g., [76], but it is rarely used in computer vision algorithm. Most often this is not a crucial omission since the main source of performance degradation is the failure of the constraint to adequately model the structure of the data.

The covariance of the noise is the product of two components in (4.2.12). The *shape* of the noise distribution is determined by the matrix C_y. This matrix is assumed to be known and can also be singular. Indeed for those variables which are available without error there is no variation along their dimensions in $\mathcal{R}^q$. The shape matrix is normalized to have $\det[\mathrm{C}_y] = 1$, where in the singular case the determinant is computed as the product of nonzero eigenvalues (which are also the singular values for a covariance matrix). For independent variables the matrix C_y is diagonal, and if all the independent variables are corrupted by the same measurement noise, $\mathrm{C}_y = \mathrm{I}_q$. This is often the case when variables of the same nature (e.g., spatial coordinates) are measured in the physical world. Note that the independency of the n measurements $\mathbf{y}_i$, and the independency of the q variables y_k are not necessarily related properties.

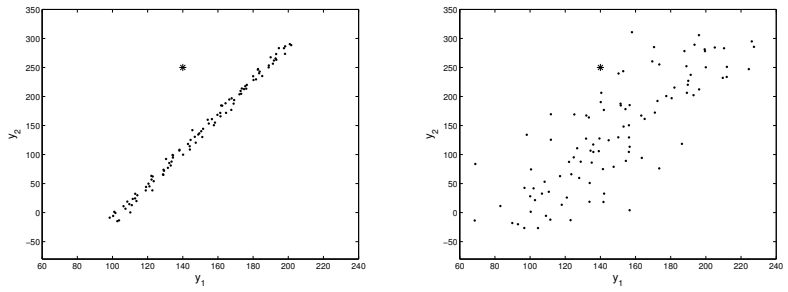
The second component of the noise covariance is the *scale* σ, which in general is not known. The main messages of this chapter will be that

> robustness in computer vision cannot be achieved without having access to a reasonably correct value of the scale.

The importance of scale is illustrated through the simple example in Figure 4.2. All the data points except the one marked with the star, belong to the same (linear) model in Figure 4.2a. The points obeying the model are called *inliers* and the point far away is an *outlier*. The shape of the noise corrupting the inliers is circular symmetric, i.e., $\sigma^2 \mathrm{C}_y = \sigma^2 \mathrm{I}_2$. The data in Figure 4.2b differs from the data in Figure 4.2a only by the value of the scale σ. Should the value of σ from the first case be used when analyzing the data in the second case, many inliers will be discarded with severe consequences on the performance of the estimation process.

The true values of the variables are not available, and instead of $\mathbf{y}_{io}$ and at the beginning of the estimation process the measurement $\mathbf{y}_i$ has to be used to compute the carriers. The first two central moments of the noise associated with a carrier can be approximated by error propagation.

Let $x_{ij} = \varphi_j(\mathbf{y}_i)$ be the j-th element, $j = 1, \ldots, p$, of the carrier vector $\mathbf{x}_i = \mathbf{x}(\mathbf{y}_i) \in \mathcal{R}^p$, computed for the i-th measurement $\mathbf{y}_i \in \mathcal{R}^q$, $i = 1, \ldots, n$. Since the measurement



(a) (b)

Figure 4.2. The importance of scale. The difference between the data in (a) and (b) is only in the scale of the noise.

vectors $\mathbf{y}_i$ are assumed to be independent, the carrier vectors $\mathbf{x}_i$ are also independent random variables.

The second order Taylor expansion of the carrier x_{ij} around the corresponding true value $x_{ijo} = \varphi_j(\mathbf{y}_{io})$ is

$$x_{ij} \approx x_{ijo} + \left[\frac{\partial \varphi_j(\mathbf{y}_{io})}{\partial \mathbf{y}}\right]^\top (\mathbf{y}_i - \mathbf{y}_{io}) + \frac{1}{2}(\mathbf{y}_i - \mathbf{y}_{io})^\top \frac{\partial^2 \varphi_j(\mathbf{y}_{io})}{\partial \mathbf{y} \partial \mathbf{y}^\top}(\mathbf{y}_i - \mathbf{y}_{io}) \quad (4.2.13)$$

where $\frac{\partial \varphi_j(\mathbf{y}_{io})}{\partial \mathbf{y}}$ is the gradient of the carrier with respect to the vector of the variables $\mathbf{y}$, and $\mathrm{H}_j(\mathbf{y}_{io}) = \frac{\partial^2 \varphi_j(\mathbf{y}_{io})}{\partial \mathbf{y} \partial \mathbf{y}^\top}$ is its Hessian matrix, both computed in the true value of the variables $\mathbf{y}_{io}$. From the measurement equation (4.2.10) and (4.2.13) the second order approximation for the expected value of the noise corrupting the carrier x_{ij} is

$$\mathrm{E}[x_{ij} - x_{ijo}] = \frac{\sigma^2}{2}\mathrm{trace}[\mathrm{C}_y \mathrm{H}_j(\mathbf{y}_{io})] \quad (4.2.14)$$

which shows that this noise is not necessarily zero-mean. The first order approximation of the noise covariance obtained by straightforward error propagation

$$\mathrm{cov}[\mathbf{x}_i - \mathbf{x}_{io}] = \sigma^2 \mathrm{C}_{x_i} = \sigma^2 \mathrm{J}_{x|y}(\mathbf{y}_{io})^\top \mathrm{C}_y \mathrm{J}_{x|y}(\mathbf{y}_{io}) \quad (4.2.15)$$

where $\mathrm{J}_{x|y}(\mathbf{y}_{io})$ is the Jacobian of the carrier vector $\mathbf{x}$ with respect to the vector of the variables $\mathbf{y}$, computed in the true values $\mathbf{y}_{io}$. In general the moments of the noise corrupting the carriers are functions of $\mathbf{y}_{io}$ and thus are point dependent. A point dependent noise process is called *heteroscedastic*. Note that the dependence is through the true values of the variables, which in general are not available. In practice, the true values are substituted with the measurements.

To illustrate the heteroscedasticity of the carrier noise we return to the example of the ellipse. From (4.2.8) we obtain the Jacobian

$$\mathrm{J}_{x|y} = \begin{bmatrix} 1 & 0 & 2y_1 & y_2 & 0 \\ 0 & 1 & 0 & y_1 & 2y_2 \end{bmatrix} \quad (4.2.16)$$

and the Hessians

$$\mathrm{H}_1 = \mathrm{H}_2 = 0 \quad \mathrm{H}_3 = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathrm{H}_4 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \mathrm{H}_5 = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix} . \qquad (4.2.17)$$

Assume that the simplest measurement noise distributed $GI(\mathbf{0}, \sigma^2 \mathrm{I}_2)$ is corrupting the two spatial coordinates (the variables). The noise corrupting the carriers, however, has nonzero mean and a covariance which is a function of $\mathbf{y}_o$

$$\mathrm{E}[\mathbf{x} - \mathbf{x}_o] = [0 \quad 0 \quad \sigma^2 \quad 0 \quad \sigma^2]^\top \qquad \mathrm{cov}[\mathbf{x} - \mathbf{x}_o] = \sigma^2 \mathrm{J}_{x|y}(\mathbf{y}_o)^\top \mathrm{J}_{x|y}(\mathbf{y}_o) \ . \quad (4.2.18)$$

To accurately estimate the parameters of the general model the heteroscedasticity of the carrier noise has to be taken into account, as will be discussed in Section 4.2.5.

4.2.2 Estimation of a Model

We can proceed now to a formal definition of the estimation process.

Given the model:

- the noisy measurements $\mathbf{y}_i$ which are the additively corrupted versions of the true values $\mathbf{y}_{io}$

$$\mathbf{y}_i = \mathbf{y}_{io} + \delta\mathbf{y}_i \quad \mathbf{y}_i \in \mathcal{R}^q \quad \delta\mathbf{y}_i \sim GI(\mathbf{0}, \sigma^2 \mathrm{C}_y) \qquad i = 1, \ldots, n$$

- the covariance of the errors $\sigma^2 \mathrm{C}_y$, known only up to the scale σ
- the constraint obeyed by the true values of the measurements

$$\alpha + \mathbf{x}_{io}^\top \boldsymbol{\theta} = 0 \quad \mathbf{x}_{io} = \mathbf{x}(\mathbf{y}_{io}) \qquad i = 1, \ldots, n$$

  and some ancillary constraints.

Find the estimates:

- for the model parameters, $\hat{\alpha}$ and $\hat{\boldsymbol{\theta}}$
- for the true values of the measurements, $\hat{\mathbf{y}}_i$
- such that they satisfy the constraint

$$\hat{\alpha} + \hat{\mathbf{x}}_i^\top \hat{\boldsymbol{\theta}} = 0 \quad \hat{\mathbf{x}}_i = \mathbf{x}(\hat{\mathbf{y}}_i) \qquad i = 1, \ldots, n$$

  and all the ancillary constraints.

The true values of the measurements $\mathbf{y}_{io}$ are called *nuisance parameters* since they have only a secondary role in the estimation process. We will treat the nuisance parameters as unknown constants, in which case we have a *functional* model [33, p.2]. When the nuisance parameters are assumed to obey a know distribution whose parameters also have to be estimated, we have a *structural* model. For robust estimation the functional models are more adequate since require less assumptions about the data.

The estimation of a functional model has two distinct parts. First, the parameter estimates are obtained in the main *parameter estimation* procedure, followed by the computation of the nuisance parameter estimates in the *data correction* procedure. The nuisance parameter estimates $\hat{\mathbf{y}}_i$ are called the *corrected data* points. The data correction procedure is usually not more than the projection of the measurements $\mathbf{y}_i$ on the already estimated constraint surface.

The parameter estimates are obtained by (most often) seeking the global minima of an *objective function*. The variables of the objective function are the normalized distances between the measurements and their true values. They are defined from the squared Mahalanobis distances

$$d_i^2 = \frac{1}{\sigma^2}(\mathbf{y}_i - \mathbf{y}_{io})^\top \mathrm{C}_y^+ (\mathbf{y}_i - \mathbf{y}_{io}) = \frac{1}{\sigma^2}\delta\mathbf{y}_i^\top \mathrm{C}_y^+ \delta\mathbf{y}_i \qquad i = 1, \ldots, n \tag{4.2.19}$$

where '+' stands for the pseudoinverse operator since the matrix C_y can be singular, in which case (4.2.19) is only a pseudodistance. Note that $d_i \geq 0$. Through the estimation procedure the $\mathbf{y}_{io}$ are replaced with $\hat{\mathbf{y}}_i$ and the distance d_i becomes the absolute value of the *normalized residual*.

The objective function $\mathcal{J}(d_1, \ldots, d_n)$ is always a positive semidefinite function taking value zero only when all the distances are zero. We should distinguish between homogeneous and nonhomogeneous objective functions. A homogeneous objective function has the property

$$\mathcal{J}(d_1, \ldots, d_n) = \frac{1}{\sigma}\mathcal{J}(\|\delta\mathbf{y}_1\|_{\mathrm{C}_y}, \ldots, \|\delta\mathbf{y}_n\|_{\mathrm{C}_y}) \tag{4.2.20}$$

where $\|\delta\mathbf{y}_i\|_{\mathrm{C}_y} = [\delta\mathbf{y}_i^\top \mathrm{C}_y^+ \delta\mathbf{y}_i]^{1/2}$ is the covariance weighted norm of the measurement error. The homogeneity of an objective function is an important property in the estimation. Only for homogeneous objective functions we have

$$[\,\hat{\alpha},\, \hat{\boldsymbol{\theta}}\,] = \underset{\alpha,\boldsymbol{\theta}}{\operatorname{argmin}}\, \mathcal{J}(d_1, \ldots, d_n) = \underset{\alpha,\boldsymbol{\theta}}{\operatorname{argmin}}\, \mathcal{J}\left(\|\delta\mathbf{y}_1\|_{\mathrm{C}_y}, \ldots, \|\delta\mathbf{y}_n\|_{\mathrm{C}_y}\right) \tag{4.2.21}$$

meaning that the scale σ does not play any role in the main estimation process. Since the value of the scale is not known a priori, by removing it an important source for performance deterioration is eliminated. All the following objective functions are homogeneous

$$\mathcal{J}_{LS} = \frac{1}{n}\sum_{i=1}^{n} d_i^2 \qquad \mathcal{J}_{LAD} = \frac{1}{n}\sum_{i=1}^{n} d_i \qquad \mathcal{J}_{LkOS} = d_{k:n} \tag{4.2.22}$$

where, $\mathcal{J}_{LS}$ yields the family of least squares estimators, $\mathcal{J}_{LAD}$ the least absolute deviations estimator, and $\mathcal{J}_{LkOS}$ the family of least k-th order statistics estimators. In an LkOS estimator the distances are assumed sorted in ascending order, and the k-th element of the list is minimized. If $k = n/2$, the least median of squares (LMedS) estimator, to be discussed in detail in Section 4.4.4, is obtained.

The most important example of nonhomogeneous objective functions is that of the M-estimators

$$\mathcal{J}_M = \frac{1}{n}\sum_{i=1}^{n} \rho(d_i) \tag{4.2.23}$$

where $\rho(u)$ is a nonnegative, even-symmetric *loss function*, nondecreasing with $|u|$. The class of $\mathcal{J}_M$ includes as particular cases $\mathcal{J}_{LS}$ and $\mathcal{J}_{LAD}$, for $\rho(u) = u^2$ and $\rho(u) = |u|$, respectively, but in general this objective function is not homogeneous. The family of M-estimators to be discussed in Section 4.4.2 have the loss function

$$\rho(u) = \begin{cases} 1 - (1 - u^2)^d & |u| \leq 1 \\ 1 & |u| > 1 \end{cases} \tag{4.2.24}$$

where $d = 0, 1, 2, 3$. It will be shown later in the chapter that all the robust techniques popular today in computer vision can be described as M-estimators.

The definitions introduced so far implicitly assumed that *all* the n data points obey the model, i.e., are inliers. In this case nonrobust estimation technique provide a satisfactory result. In the presence of outliers, only $n_1 \leq n$ measurements are inliers and obey (4.2.3). The number n_1 is not know. The measurement equation (4.2.10) becomes

$$\begin{aligned} &\mathbf{y}_i = \mathbf{y}_{io} + \delta\mathbf{y}_i \qquad \delta\mathbf{y}_i \sim GI(\mathbf{0}, \sigma^2 \mathrm{C}_y) \qquad i = 1, \ldots, n_1 \\ &\mathbf{y}_i \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad i = (n_1 + 1), \ldots, n \end{aligned} \tag{4.2.25}$$

where nothing is assumed known about the $n - n_1$ outliers. Sometimes in robust methods proposed in computer vision, such as [100], [107], [114], the outliers were modeled as obeying a uniform distribution.

A robust method has to determine n_1 simultaneously with the estimation of the inlier model parameters. Since n_1 is unknown, at the beginning of the estimation process the model is still defined for $i = 1, \ldots, n$. Only through the optimization of an adequate objective function are the data points classified into inliers or outliers. The result of the robust estimation is the *inlier/outlier dichotomy* of the data.

The estimation process maps the input, the set of measurements $\mathbf{y}_i$, $i = 1, \ldots, n$ into the output, the estimates $\hat{\alpha}$, $\hat{\boldsymbol{\theta}}$ and $\hat{\mathbf{y}}_i$. The measurements are noisy and the uncertainty about their true value is mapped into the uncertainty about the true value of the estimates. The computational procedure employed to obtain the estimates is called the *estimator*. To describe the properties of an estimator the estimates are treated as random variables. The estimate $\hat{\boldsymbol{\theta}}$ will be used generically in the next two sections to discuss these properties.

4.2.3 Robustness of an Estimator

Depending on n, the number of available measurements, we should distinguish between small (finite) sample and large (asymptotic) sample properties of an estimator [76, Secs.6,7]. In the latter case n becomes large enough that further increase in its value no longer has a significant influence on the estimates. Many of the estimator properties proven in theoretical statistics are asymptotic, and are not necessarily valid for small data sets. Rigorous analysis of small sample properties is difficult. See [86] for examples in pattern recognition.

What is a small or a large sample depends on the estimation problem at hand. Whenever the model is not accurate even for a large number of measurements the estimate remains highly sensitive to the input. This situation is frequently present in computer vision, where

only a few tasks would qualify as large scale behavior of the employed estimator. We will not discuss here asymptotic properties, such as the *consistency*, which describes the relation of the estimate to its true value when the number of data points grows unbounded. Our focus is on the *bias* of an estimator, the property which is also central in establishing whether the estimator is robust or not.

Let $\boldsymbol{\theta}$ be the true value of the estimate $\hat{\boldsymbol{\theta}}$. The estimator mapping the measurements $\mathbf{y}_i$ into $\hat{\boldsymbol{\theta}}$ is unbiased if

$$\mathrm{E}[\hat{\boldsymbol{\theta}}] = \boldsymbol{\theta} \tag{4.2.26}$$

where the expectation is taken over all possible sets of measurements of size n, i.e., over the joint distribution of the q variables. Assume now that the input data contains n_1 inliers and $n - n_1$ outliers. In a "thought" experiment we keep all the inliers fixed and allow the outliers to be placed anywhere in $\mathcal{R}^q$, the space of the measurements $\mathbf{y}_i$. Clearly, some of these arrangements will have a larger effect on $\hat{\boldsymbol{\theta}}$ than others. Will define the *maximum bias* as

$$b_{max}(n_1, n) = \max_{\mathcal{O}} \|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\| \tag{4.2.27}$$

where $\mathcal{O}$ stands for the arrangements of the $n - n_1$ outliers. Will say that an estimator exhibits a *globally robust* behavior in a given task if and only if

$$\text{for} \quad n_1 < n \qquad b_{max}(n_1, n) < t_b \tag{4.2.28}$$

where $t_b \geq 0$ is a threshold depending on the task. That is, the presence of outliers cannot introduce an estimation error beyond the tolerance deemed acceptable for that task. To *qualitatively* assess the robustness of the estimator we can define

$$\eta(n) = 1 - \min_{n_1} \frac{n_1}{n} \qquad \text{while (4.2.28) holds} \tag{4.2.29}$$

which measures its outlier rejection capability. Note that the definition is based on the worst case situation which may not appear in practice.

The robustness of an estimator is assured by the employed objective function. Among the three homogeneous objective functions in (4.2.22), minimization of two criteria, the least squares $\mathcal{J}_{LS}$ and the least absolute deviations $\mathcal{J}_{LAD}$, does not yield robust estimators. A striking example for the (less known) nonrobustness of the latter is discussed in [90, p.20]. The LS and LAD estimators are not robust since their homogeneous objective function (4.2.20) is also *symmetric*. The value of a symmetric function is invariant under the permutations of its variables, the distances d_i in our case. Thus, in a symmetric function all the variables have equal importance.

To understand why these two objective functions lead to a nonrobust estimator, consider the data containing a single outlier located far away from all the other points, the inliers (Figure 4.2a). The scale of the inlier noise, σ, has no bearing on the minimization of a homogeneous objective function (4.2.21). The symmetry of the objective function, on the other hand, implies that during the optimization *all* the data points, including the outlier, are treated in the same way. For a parameter estimate close to the true value the outlier yields a very large measurement error $\|\delta\mathbf{y}_i\|_{\mathrm{C}_y}$. The optimization procedure therefore tries to

compensate for this error and biases the fit toward the outlier. For any threshold t_b on the tolerated estimation errors, the outlier can be placed far enough from the inliers such that (4.2.28) is not satisfied. This means $\eta(n) = 0$.

In a robust technique the objective function cannot be both symmetric and homogeneous. For the M-estimators $\mathcal{J}_M$ (4.2.23) is only symmetric, while the least k-th order statistics objective function $\mathcal{J}_{LkOS}$ (4.2.20) is only homogeneous.

Consider $\mathcal{J}_{LkOS}$. When at least k measurements in the data are inliers and the parameter estimate is close to the true value, the k-th error is computed based on an inlier and it is small. The influence of the outliers is avoided, and if (4.2.28) is satisfied, for the LkOS estimator $\eta(n) = (n - k)/n$. As will be shown in the next section, the condition (4.2.28) depends on the level of noise corrupting the inliers. When the noise is large, the value of $\eta(n)$ decreases. Therefore, it is important to realize that $\eta(n)$ only measures the global robustness of the employed estimator in the context of the task. However, this is what we really care about in an application!

Several strategies can be adopted to define the value of k. Prior to the estimation process k can be set to a given percentage of the number of points n. For example, if $k = n/2$ the least median of squares (LMedS) estimator [87] is obtained. Similarly, the value of k can be defined implicitly by setting the level of the allowed measurement noise and maximizing the number of data points within this tolerance. This is the approach used in the random sample consensus (RANSAC) estimator [26] which solves

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \max_{k} \; d_{k:n} \qquad \text{subject to} \quad \|\delta \mathbf{y}_{k:n}\|_{\mathrm{C}_y} < s(\sigma) \tag{4.2.30}$$

where $s(\sigma)$ is a *user set threshold* related to the scale of the inlier noise. In a third, less generic strategy, an auxiliary optimization process is introduced to determine the best value of k by analyzing a sequence of scale estimates [63], [77].

Beside the global robustness property discussed until now, the *local robustness* of an estimator also has to be considered when evaluating performance. Local robustness is measured through the *gross error sensitivity* which describes the worst influence a single measurement can have on the value of the estimate [90, p.191]. Local robustness is a central concept in the theoretical analysis of robust estimators and has a complex relation to global robustness e.g., [40], [69]. It also has important practical implications.

Large gross error sensitivity (poor local robustness) means that for a critical arrangement of the n data points, a slight change in the value of a measurement $\mathbf{y}_i$ yields an unexpectedly large change in the value of the estimate $\hat{\boldsymbol{\theta}}$. Such behavior is certainly undesirable. Several robust estimators in computer vision, such as LMedS and RANSAC have large gross error sensitivity, as will be shown in Section 4.4.4.

4.2.4 Definition of Robustness

We have defined global robustness in a task specific manner. An estimator is considered robust only when the estimation error is guaranteed to be less than what can be tolerated in the application (4.2.28). This definition is different from the one used in statistic, where global robustness is closely related to the *breakdown point* of an estimator. The (explosion)

breakdown point is the minimum percentage of outliers in the data for which the value of maximum bias *becomes unbounded* [90, p.117]. Also, the maximum bias is defined in statistics relative to a *typically good* estimate computed with all the points being inliers, and not relative to the true value as in (4.2.27).

For computer vision problems the statistical definition of robustness is too narrow. First, a finite maximum bias can still imply unacceptably large estimation errors. Second, in statistics the estimators of models linear in the variables are often required to be affine equivariant, i.e., an affine transformation of the input (measurements) should change the output (estimates) by the inverse of that transformation [90, p.116]. It can be shown that the breakdown point of an affine equivariant estimator cannot exceed 0.5, i.e., the inliers must be the absolute majority in the data [90, p.253], [69]. According to the definition of robustness in statistics, once the number of outliers exceeds that of inliers, the former can be arranged into a false structure thus compromising the estimation process.

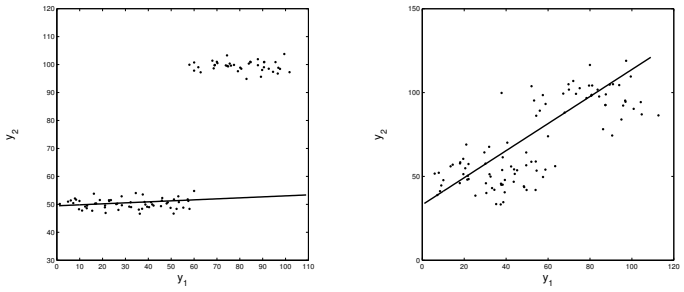
Our definition of robust behavior is better suited for estimation in computer vision where often the information of interest is carried by less than half of the data points and/or the data may also contain multiple structures. Data with multiple structures is characterized by the presence of several instances of the *same* model, each corresponding in (4.2.11) to a different set of parameters $\alpha_k, \boldsymbol{\theta}_k$, $k = 1, \ldots, K$. Independently moving objects in a scene is just one example in which such data can appear. (The case of simultaneous presence of different models is too rare to be considered here.)

The data in Figure 4.3 is a simple example of the multistructured case. Outliers not belonging to any of the model instances can also be present. During the estimation of any of the individual structures, all the other data points act as outliers. Multistructured data is very challenging and once the measurement noise becomes large (Figure 4.3b) none of the current robust estimators can handle it. Theoretical analysis of robust processing for data containing two structures can be found in [9], [98], and we will discuss it in Section 4.4.7.

The definition of robustness employed here, beside being better suited for data in computer vision, also has the advantage of highlighting the complex relation between σ, the scale of the inlier noise and $\eta(n)$, the amount of outlier tolerance. To avoid misconceptions we do not recommend the use of the term breakdown point in the context of computer vision.

Assume for the moment that the data contains only inliers. Since the input is corrupted by measurement noise, the estimate $\hat{\boldsymbol{\theta}}$ will differ from the true value $\boldsymbol{\theta}$. The larger the scale of the inlier noise, the higher the probability of a significant deviation between $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$. The inherent uncertainty of an estimate computed from noisy data thus sets a lower bound on t_b (4.2.28). Several such bounds can be defined, the best known being the Cramer-Rao bound [76, p.78]. Most bounds are computed under strict assumptions about the distribution of the measurement noise. Given the complexity of the visual data, the significance of a bound in a real application is often questionable. For a discussion of the Cramer-Rao bound in the context of computer vision see [58, Chap. 14], and for an example [96].

Next, assume that the employed robust method can handle the percentage of outliers present in the data. After the outliers were removed, the estimate $\hat{\boldsymbol{\theta}}$ is computed from less data points and therefore it is less reliable (a small sample property). The probability of a larger deviation from the true value increases, which is equivalent to an increase of the



(a) (b)

Figure 4.3. Multistructured data. The measurement noise is small in (a) and large in (b). The line is the fit obtained with the least median of squares (LMedS) estimator.

lower bound on t_b. Thus, for a given level of the measurement noise (the value of σ), as the employed estimator has to remove more outliers from the data, the chance of larger estimation errors (the lower bound on t_b) also increases. The same effect is obtained when the number of removed outliers is kept the same but the level of the measurement noise increases.

In practice, the tolerance threshold t_b is set by the application to be solved. When the level of the measurement noise corrupting the inliers increases, eventually we are no longer able to keep the estimation errors below t_b. Based on our definition of robustness the estimator no longer can be considered as being robust! Note that by defining robustness through the breakdown point, as it is done in statistics, the failure of the estimator would not have been recorded. Our definition of robustness also covers the *numerical robustness* of a nonrobust estimator when all the data obeys the model. In this case the focus is exclusively on the size of the estimation errors, and the property is related to the *efficiency* of the estimator.

The loss of robustness is best illustrated with multistructured data. For example, the LMedS estimator was designed to reject up to half the points being outliers. When used to robustly fit a line to the data in Figure 4.3a, correctly recovers the lower structure which contains sixty percent of the points. However, when applied to the similar but heavily corrupted data in Figure 4.3b, LMedS completely fails and the obtained fit is not different from that of the nonrobust least squares [9], [78], [98]. As will be shown in Section 4.4.7, the failure of LMedS is part of a more general deficiency of robust estimators.

4.2.5 Taxonomy of Estimation Problems

The model described at the beginning of Section 4.2.2, the measurement equation

$$\mathbf{y}_i = \mathbf{y}_{io} + \delta\mathbf{y}_i \quad \mathbf{y}_i \in \mathcal{R}^q \quad \delta\mathbf{y}_i \sim GI(\mathbf{0}, \sigma^2 \mathrm{C}_y) \qquad i = 1, \ldots, n \tag{4.2.31}$$

and the constraint

$$\alpha + \mathbf{x}_{io}^\top \boldsymbol{\theta} = 0 \quad \mathbf{x}_{io} = \mathbf{x}(\mathbf{y}_{io}) \qquad i = 1, \ldots, n \tag{4.2.32}$$

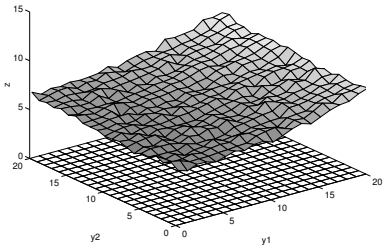


Figure 4.4. A typical traditional regression problem. Estimate the parameters of the surface defined on a sampling grid.

is general enough to apply to almost all computer vision problems. The constraint is linear in the parameters α and $\boldsymbol{\theta}$, but nonlinear in the variables $\mathbf{y}_i$. A model in which all the variables are measured with errors is called in statistics an *errors-in-variables* (EIV) model [112], [116].

We have already discussed in Section 4.2.1 the problem of ellipse fitting using such nonlinear EIV model (Figure 4.1). Nonlinear EIV models also appear in any computer vision problem in which the constraint has to capture an incidence relation in projective geometry. For example, consider the epipolar constraint between the affine coordinates of corresponding points in two images A and B

$$[y_{B1o} \quad y_{B2o} \quad 1]^\top \mathrm{F} \, [y_{A1o} \quad y_{A2o} \quad 1] = 0 \tag{4.2.33}$$

where F is a rank two matrix [43, Chap.8]. When this bilinear constraint is rewritten as (4.2.32) four of the eight carriers

$$\mathbf{x}_o^\top = [y_{A1o} \quad y_{A2o} \quad y_{B1o} \quad y_{B2o} \quad y_{A1o}y_{B1o} \quad y_{A2o}y_{B1o} \quad y_{A2o}y_{B1o} \quad y_{A2o}y_{B2o}] \tag{4.2.34}$$

are nonlinear functions in the variables. Several nonlinear EIV models used in recovering 3D structure from uncalibrated image sequences are discussed in [34].

To obtain an unbiased estimate, the parameters of a nonlinear EIV model have to be computed with nonlinear optimization techniques such as the Levenberg-Marquardt method. See [43, Appen.4] for a discussion. However, the estimation problem can be also approached as a *linear model in the carriers* and taking into account the heteroscedasticity of the noise process associated with the carriers (Section 4.2.1). Several such techniques were proposed in the computer vision literature: the renormalization method [58], the heteroscedastic errors-in-variables (HEIV) estimator [64], [71], [70] and the fundamental numerical scheme (FNS) [12]. All of them return estimates unbiased in a first order approximation.

Since the focus of this chapter is on robust estimation, we will only use the less general, *linear errors-in-variables regression* model. In this case the carriers are linear expressions in the variables, and the constraint (4.2.32) becomes

$$\alpha + \mathbf{y}_{io}^\top \boldsymbol{\theta} = 0 \qquad i = 1, \ldots, n \, . \tag{4.2.35}$$

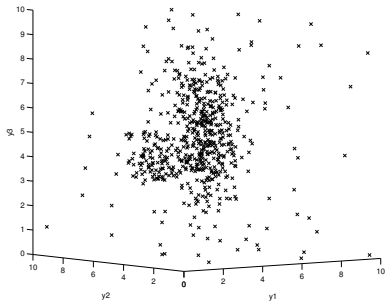


Figure 4.5. A typical location problem. Determine the center of the cluster.

An important particular case of the general EIV model is obtained by considering the constraint (4.2.4). This is the *traditional regression* model where only a single variable, denoted z, is measured with error and therefore the measurement equation becomes

$$\begin{aligned} z_i &= z_{io} + \delta z_i \qquad \delta z_i \sim GI(0,\sigma^2) \qquad i = 1,\dots,n \\ \mathbf{y}_i &= \mathbf{y}_{io} \qquad\qquad\qquad\qquad\quad i = 1,\dots,n \end{aligned} \tag{4.2.36}$$

while the constraint is expressed as

$$z_{io} = \alpha + \mathbf{x}_{io}^\top \boldsymbol{\theta} \qquad \mathbf{x}_{io} = \mathbf{x}(\mathbf{y}_{io}) \qquad i = 1,\dots,n\,. \tag{4.2.37}$$

Note that the nonlinearity of the carriers is no longer relevant in the traditional regression model since now their value is known.

In traditional regression the covariance matrix of the variable vector

$$\mathbf{v}^\top = [z \quad \mathbf{y}] \qquad\qquad \sigma^2 \mathrm{C}_v = \sigma^2 \begin{bmatrix} 1 & \mathbf{0}^\top \\ \mathbf{0} & \mathrm{O} \end{bmatrix} \tag{4.2.38}$$

has rank one, and the normalized distances, d_i (4.2.19) used in the objective functions become

$$d_i^2 = \frac{1}{\sigma^2}(\mathbf{v}_i - \mathbf{v}_{io})^\top \mathrm{C}_v^+ (\mathbf{v}_i - \mathbf{v}_{io}) = \frac{(z_i - z_{io})^2}{\sigma^2} = \left(\frac{\delta z_i}{\sigma}\right)^2 . \tag{4.2.39}$$

The two regression models, the linear EIV (4.2.35) and the traditional (4.2.37), has to be estimated with different least squares techniques, as will be shown in Section 4.4.1. Using the method optimal for traditional regression when estimating an EIV regression model, yields biased estimates. In computer vision the traditional regression model appears almost exclusively only when an image defined on the sampling grid is to be processed. In this case the pixel coordinates are the independent variables and can be considered available uncorrupted (Figure 4.4).

All the models discussed so far were related the class of *regression problems*. A second, equally important class of estimation problems also exist. They are the *location problems* in

which the goal is to determine an estimate for the "center" of a set of noisy measurements. The location problems are closely related to clustering in pattern recognition.

In practice a location problem is of interest only in the context of robust estimation. The measurement equation is

$$\begin{aligned} &\mathbf{y}_i = \mathbf{y}_{io} + \delta\mathbf{y}_i && i = 1, \dots, n_1 \\ &\mathbf{y}_i && i = (n_1 + 1), \dots, n \end{aligned} \tag{4.2.40}$$

with the constraint

$$\mathbf{y}_{io} = \boldsymbol{\theta} \qquad i = 1, \dots, n_1 \tag{4.2.41}$$

with n_1, the number of inliers, unknown.

The important difference from the regression case (4.2.25) is that now we do not assume that the noise corrupting the inliers can be characterized by a single covariance matrix, i.e., the cloud of inliers has an elliptical shape. This will allow to handle data such as in Figure 4.5.

The goal of the estimation process in a location problem is twofold.

- Find a robust estimate $\hat{\boldsymbol{\theta}}$ for the center of the n measurements.
- Select the n_1 data points associated with this center.

The discussion in Section 4.2.4 about the definition of robustness also applies to location estimation.

While handling multistructured data in regression problems is an open research question, clustering multistructured data is the main application of the location estimators. The feature spaces derived from visual data are complex and usually contain several clusters. The goal of feature space analysis is to delineate each significant cluster through a robust location estimation process. We will return to location estimation in Section 4.3.

4.2.6 Linear Errors-in-Variables Regression Model

To focus on the issue of robustness in regression problems, only the simplest linear errors-in-variables (EIV) regression model (4.2.35) will be used. The measurements are corrupted by i.i.d. noise

$$\mathbf{y}_i = \mathbf{y}_{io} + \delta\mathbf{y}_i \quad \mathbf{y}_i \in \mathcal{R}^p \quad \delta\mathbf{y}_i \sim GI(\mathbf{0}, \sigma^2 \mathrm{I}_p) \qquad i = 1, \dots, n \tag{4.2.42}$$

where the number of variables q was aligned with p the dimension of the parameter vector $\boldsymbol{\theta}$. The constraint is rewritten under the more convenient form

$$g(\mathbf{y}_{io}) = \mathbf{y}_{io}^\top \boldsymbol{\theta} - \alpha = 0 \qquad i = 1, \dots, n\,. \tag{4.2.43}$$

To eliminate the ambiguity up to a constant of the parameters the following two ancillary constraints are used

$$\|\boldsymbol{\theta}\| = 1 \qquad \alpha \geq 0\,. \tag{4.2.44}$$

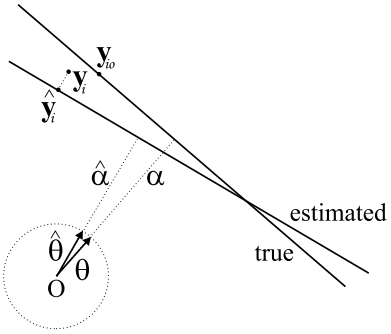


Figure 4.6. The concepts of the linear errors-in-variables regression model. The constraint is in the Hessian normal form.

The three constraints together define the Hessian normal form of a plane in $\mathcal{R}^p$. Figure 4.6 shows the interpretation of the two parameters. The unit vector $\boldsymbol{\theta}$ is the direction of the normal, while α is the distance from the origin.

In general, given a surface $f(\mathbf{y}_o) = 0$ in $\mathcal{R}^p$, the first order approximation of the shortest Euclidean distance from a point $\mathbf{y}$ to the surface is [111, p.101]

$$\|\mathbf{y} - \hat{\mathbf{y}}\| \simeq \frac{|f(\mathbf{y})|}{\|\nabla f(\hat{\mathbf{y}})\|} \tag{4.2.45}$$

where $\hat{\mathbf{y}}$ is the orthogonal projection of the point onto the surface, and $\nabla f(\hat{\mathbf{y}})$ is the gradient computed in the location of that projection. The quantity $f(\mathbf{y})$ is called the *algebraic distance*, and it can be shown that it is zero only when $\mathbf{y} = \mathbf{y}_o$, i.e., the point is on the surface.

Taking into account the linearity of the constraint (4.2.43) and that $\boldsymbol{\theta}$ has unit norm, (4.2.45) becomes

$$\|\mathbf{y} - \hat{\mathbf{y}}\| = |g(\mathbf{y})| \tag{4.2.46}$$

i.e., the Euclidean distance from a point to a hyperplane written under the Hessian normal form is the absolute value of the algebraic distance.

When all the data points obey the model the least squares objective function $\mathcal{J}_{LS}$ (4.2.22) is used to estimate the parameters of the linear EIV regression model. The i.i.d. measurement noise (4.2.42) simplifies the expression of the distances d_i (4.2.19) and the minimization problem (4.2.21) can be written as

$$[\,\hat{\alpha},\, \hat{\boldsymbol{\theta}}\,] = \underset{\alpha,\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{y}_i - \mathbf{y}_{io}\|^2 \,. \tag{4.2.47}$$

Combining (4.2.46) and (4.2.47) we obtain

$$[\,\hat{\alpha},\, \hat{\boldsymbol{\theta}}\,] = \underset{\alpha,\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} g(\mathbf{y}_i)^2 \,. \tag{4.2.48}$$

To solve (4.2.48) the true values $\mathbf{y}_{io}$ are replaced with the orthogonal projection of the $\mathbf{y}_i$-s onto the hyperplane. The orthogonal projections $\hat{\mathbf{y}}_i$ associated with the solution $\hat{\alpha}, \hat{\boldsymbol{\theta}}$ are the corrected values of the measurements $\mathbf{y}_i$, and satisfy (Figure 4.6)

$$\hat{g}(\hat{\mathbf{y}}_i) = \hat{\mathbf{y}}_i^\top \hat{\boldsymbol{\theta}} - \hat{\alpha} = 0 \qquad i = 1, \ldots, n \; . \tag{4.2.49}$$

The estimation process (to be discussed in Section 4.4.1) returns the parameter estimates, after which the ancillary constrains (4.2.44) can be imposed. The employed parametrization of the linear model

$$\boldsymbol{\omega}_1 = [\boldsymbol{\theta}^\top \; \alpha]^\top = [\theta_1 \; \theta_2 \; \cdots \; \theta_p \; \; \alpha]^\top \in \mathcal{R}^{p+1} \tag{4.2.50}$$

however is redundant. The vector $\boldsymbol{\theta}$ being a unit vector it is restricted to the p-dimensional unit sphere in R^p. This can be taken into account by expressing $\boldsymbol{\theta}$ in polar angles [116] as, $\boldsymbol{\theta} = \boldsymbol{\theta}(\boldsymbol{\beta})$

$$\boldsymbol{\beta} = [\beta_1 \; \beta_2 \; \cdots \; \beta_{p-1}]^\top \quad 0 \le \beta_j \le \pi \;\; j = 1, \cdots, p-2 \qquad 0 \le \beta_{p-1} < 2\pi \tag{4.2.51}$$

where the mapping is

$$\begin{aligned}
\theta_1(\boldsymbol{\beta}) &= sin\beta_1 \cdots sin\beta_{p-2} \, sin\beta_{p-1} \\
\theta_2(\boldsymbol{\beta}) &= sin\beta_1 \cdots sin\beta_{p-2} \, cos\beta_{p-1} \\
\theta_3(\boldsymbol{\beta}) &= sin\beta_1 \cdots sin\beta_{p-3} \, cos\beta_{p-2} \\
&\vdots \\
\theta_{p-1}(\boldsymbol{\beta}) &= sin\beta_1 \, cos\beta_2 \\
\theta_p(\boldsymbol{\beta}) &= cos\beta_1 \; .
\end{aligned} \tag{4.2.52}$$

The polar angles β_j and α provide the second representation of a hyperplane

$$\boldsymbol{\omega}_2 = [\boldsymbol{\beta}^\top \; \alpha]^\top = [\beta_1 \; \beta_2 \; \cdots \; \beta_{p-1} \; \; \alpha]^\top \in \mathcal{R}^p \; . \tag{4.2.53}$$

The $\boldsymbol{\omega}_2$ representation being based in part on the mapping from the unit sphere to $\mathcal{R}^{p-1}$, is inherently discontinuos. See [24, Chap. 5] for a detailed discussion of such representations. The problem is well known in the context of Hough transform, where this parametrization is widely used.

To illustrate the discontinuity of the mapping, consider the representation of a line, $p = 2$. In this case only a single polar angle β is needed and the equation of a line in the Hessian normal form is

$$y_1 cos\beta + y_2 sin\beta - \alpha = 0 \; . \tag{4.2.54}$$

In Figures 4.7a and 4.7b two pairs of lines are shown, each pair having the same α but different polar angles. Take $\beta_1 = \beta$. The lines in Figure 4.7a have the relation $\beta_2 = \beta + \pi$, while those in Figure 4.7b $\beta_2 = 2\pi - \beta$. When represented in the $\boldsymbol{\omega}_2$ parameter space (Figure 4.7c) the four lines are mapped into four points.

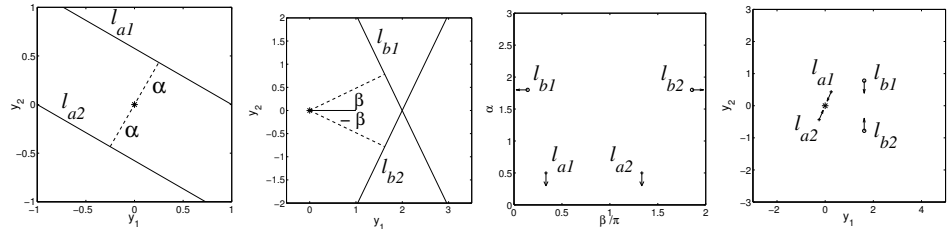


Figure 4.7. Discontinuous mappings due to the polar representation of $\boldsymbol{\theta}$. (a) Two lines with same α and antipodal polar angles β. (b) Two lines with same α and polar angles β differing only in sign. (c) The $\boldsymbol{\omega}_2$ parameter space. (d) The $\boldsymbol{\omega}_3$ parameter space.

Let now $\alpha \to 0$ for the first pair, and $\beta \to 0$ for the second pair. In the input space each pair of lines merges into a single line, but the four points in the $\boldsymbol{\omega}_2$ parameter space remain distinct, as shown by the arrows in Figure 4.7c.

A different parameterization of the hyperplane in $\mathcal{R}^p$ can avoid this problem, though no representation of the Hessian normal form can provide a continuous mapping into a feature space. In the new parameterization all the hyperplanes not passing through the origin are represented by their point closest to the origin. This point has the coordinates $\alpha\boldsymbol{\theta}$ and is the intersection of the plane with the normal from the origin. The new parametrization is

$$\boldsymbol{\omega}_3 = \alpha\boldsymbol{\theta} = [\alpha\theta_1 \; \alpha\theta_2 \; \cdots \; \alpha\theta_p]^\top \in \mathcal{R}^p \; . \tag{4.2.55}$$

It is important to notice that the space of $\boldsymbol{\omega}_3$ is in fact the space of the input as can also be seen from Figure 4.6. Thus, when the pairs of lines collapse, so do their representations in the $\boldsymbol{\omega}_3$ space (Figure 4.7d).

Planes which contain the origin have to be treated separately. In practice this also applies to planes passing near the origin. A plane with small α is translated along the direction of the normal $\boldsymbol{\theta}$ with a known quantity τ. The plane is then represented as $\tau\boldsymbol{\theta}$. When m planes are close to the origin the direction of translation is $\sum_{i=1}^{m} \boldsymbol{\theta}_i$ and the parameters of each translated plane are adjusted accordingly. After processing in the $\boldsymbol{\omega}_3$ space it is easy to convert back to the $\boldsymbol{\omega}_1$ representation.

Estimation of the linear EIV regression model parameters by total least squares (Section 4.4.1) uses the $\boldsymbol{\omega}_1$ parametrization. The $\boldsymbol{\omega}_2$ parametrization will be employed in the robust estimation of the model (Section 4.4.5). The parametrization $\boldsymbol{\omega}_3$ is useful when the problem of robust multiple regression is approached as a feature space analysis problem [9].

4.2.7 Objective Function Optimization

The objective functions used in robust estimation are often nondifferentiable and analytical optimization methods, like those based on the gradient, cannot be employed. The k-th order statistics, $\mathcal{J}_{LkOS}$ (4.2.22) is such an objective function. Nondifferentiable objective

functions also have many local extrema and to avoid being trapped in one of these minima the optimization procedure should be run starting from several initial positions. A numerical technique to implement robust estimators with nondifferentiable objective functions, is based on *elemental subsets*.

An elemental subset is the smallest number of data points required to fully instantiate a model. In the linear EIV regression case this means p points in a general position, i.e., the points define a basis for a $(p-1)$-dimensional affine subspace in $\mathcal{R}^p$ [90, p. 257]. For example, if $p = 3$ not all three points can lie on a line in 3D.

The p points in an elemental subset thus define a full rank system of equations from which the model parameters α and $\boldsymbol{\theta}$ can be computed analytically. Note that using p points suffices to solve this homogeneous system. The ancillary constraint $\|\boldsymbol{\theta}\| = 1$ is imposed at the end. The obtained parameter vector $\boldsymbol{\omega}_1 = [\boldsymbol{\theta}^\top \ \alpha]^\top$ will be called, with a slight abuse of notation, a *model candidate*.

The number of possibly distinct elemental subsets in the data $\begin{pmatrix} n \\ p \end{pmatrix}$, can be very large. In practice an exhaustive search over all the elemental subsets is not feasible, and a *random sampling* of this ensemble has to be used. The sampling drastically reduces the amount of computations at the price of a negligible decrease in the outlier rejection capability of the implemented robust estimator.

Assume that the number of inliers in the data is n_1, and that N elemental subsets, p-tuples, were drawn independently from that data. The probability that *none* of these subsets contains only inliers is (after disregarding the artifacts due to the finite sample size)

$$P_{fail} = \left[1 - \left(\frac{n_1}{n}\right)^p\right]^N \ . \tag{4.2.56}$$

We can choose a small probability P_{error} to bound upward P_{fail}. Then the equation

$$P_{fail} = P_{error} \tag{4.2.57}$$

provides the value of N as a function of the percentage of inliers n_1/n, the dimension of the parameter space p and P_{error}. This probabilistic sampling strategy was applied independently in computer vision for the RANSAC estimator [26] and in statistics for the LMedS estimator [90, p.198].

Several important observations has to be made. The value of N obtained from (4.2.57) is an absolute lower bound since it implies that *any* elemental subset which contains only inliers can provide a satisfactory model candidate. However, the model candidates are computed from the smallest possible number of data points and the influence of the noise is the largest possible. Thus, the assumption used to compute N is not guaranteed to be satisfied once the measurement noise becomes significant. In practice n_1 is not know prior to the estimation, and the value of N has to be chosen large enough to compensate for the inlier noise under a worst case scenario.

Nevertheless, it is not recommended to increase the size of the subsets. The reason is immediately revealed if we define in a drawing of subsets of size $q \geq p$, the probability of

success as obtaining a subset which contains only inliers

$$P_{success} = \frac{\begin{pmatrix} n_1 \\ q \end{pmatrix}}{\begin{pmatrix} n \\ q \end{pmatrix}} = \prod_{k=0}^{q-1} \frac{n_1 - k}{n - k} \,. \tag{4.2.58}$$

This probability is maximized when $q = p$.

Optimization of an objective function using random elemental subsets is *only a computational tool* and has no bearing on the robustness of the corresponding estimator. This fact is not always recognized in the computer vision literature. However, *any* estimator can be implemented using the following numerical optimization procedure.

Objective Function Optimization With Elemental Subsets

- Repeat N times:
  1. choose an elemental subset (p-tuple) by random sampling;
  2. compute the corresponding model candidate;
  3. compute the value of the objective function by assuming the model candidate valid for all the data points.
- The parameter estimate is the model candidate yielding the smallest (largest) objective function value.

This procedure can be applied the same way for the nonrobust least squares objective function $\mathcal{J}_{LS}$ as for the the robust least k-th order statistics $\mathcal{J}_{LkOS}$ (4.2.22). However, while an analytical solution is available for the former (Section 4.4.1), for the latter the above procedure is the only practical way to obtain the estimates.

Performing an exhaustive search over all elemental subsets does not guarantee to find the global extremum of the objective function since not every location in the parameter space can be visited. Finding the global extremum, however, most often is also not required. When a robust estimator is implemented with the elemental subsets based search procedure, the goal is only to obtain the inlier/outlier dichotomy, i.e., to select the "good" data. The robust estimate corresponding to an elemental subset is then refined by processing the selected inliers with a nonrobust (least squares) estimator. See [88] for an extensive discussion of the related issues from a statistical perspective.

The number of required elemental subsets N can be significantly reduced when information about the reliability of the data points is available. This information can be either provided by the user, or can be derived from the data through an auxiliary estimation process. The elemental subsets are then chosen with a *guided sampling* biased toward the points having a higher probability to be inliers. See [104] and [105] for computer vision examples.

We have emphasized that the random sampling of elemental subsets is not more than a computational procedure. However, guided sampling has a different nature since it relies on a fuzzy pre-classification of the data (derived automatically, or supplied by the user). Guided sampling can yield a significant improvement in the performance of the estimator

relative to the unguided approach. The better quality of the elemental subsets can be converted into either less samples in the numerical optimization (while preserving the outlier rejection capacity $\eta(n)$ of the estimator), or into an increase of $\eta(n)$ (while preserving the same number of elemental subsets N).

We conclude that guided sampling should be regarded as a robust technique, while the random sampling procedure should be not. Their subtle but important difference has to be recognized when designing robust methods for solving complex vision tasks.

In most applications information reliable enough to guide the sampling is not available. However, the amount of computations still can be reduced by performing in the space of the parameters local searches with optimization techniques which do not rely on derivatives. For example, in [91] line search was proposed to improve the implementation of the LMedS estimator. Let $\boldsymbol{\omega}_1^b = [\boldsymbol{\theta}_b^\top \ \alpha_b]^\top$ be the currently best model candidate, as measured by the value of the objective function. From the next elemental subset the model candidate $\boldsymbol{\omega}_1 = [\boldsymbol{\theta}^\top \ \alpha]^\top$ is computed. The objective function is then assessed at several locations along the line segment $\boldsymbol{\omega}_1^b - \boldsymbol{\omega}_1$, and if an improvement relative to $\boldsymbol{\omega}_1^b$ is obtained the best model candidate is updated.

In Section 4.4.5 we will use a more effective multidimensional unconstrained optimization technique, the *simplex based direct search*. The simplex search is a heuristic method proposed in 1965 by Nelder and Mead [79]. See also [83, Sec.10.4]. Simplex search is a heuristic with no theoretical foundations. Recently direct search methods became again of interest and significant progress was reported in the literature [66] [115], but in our context there is no need to use these computationally more intensive techniques.

To take into account the fact that $\boldsymbol{\theta}$ is a unit vector, the simplex search should be performed in the space of the polar angles $\boldsymbol{\beta} \in R^{p-1}$. A simplex in R^{p-1} is the volume delineated by p vertices in a nondegenerate position, i.e., the points define an affine basis in R^{p-1}. For example, in R^2 the simplex is a triangle, in R^3 it is a tetrahedron. In our case, the vertices of the simplex are the polar angle vectors $\boldsymbol{\beta}_k \in R^{p-1}$, $k = 1, \ldots, p$, representing p unit vectors $\boldsymbol{\theta}_k \in R^p$. Each vertex is associated with the value of a scalar function $f_k = f(\boldsymbol{\beta}_k)$. For example, $f(u)$ can be the objective function of an estimator. The goal of the search is to find the global (say) maximum of this function.

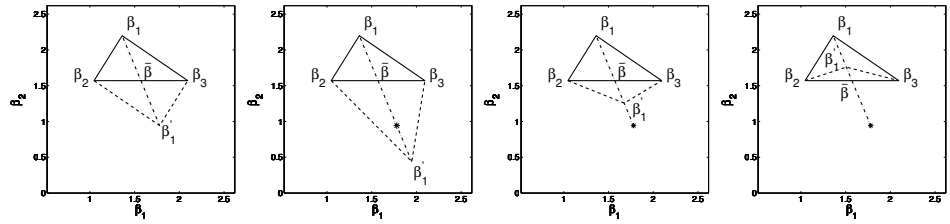
We can always assume that at the beginning of an iteration the vertices are labeled such that $f_1 \leq f_2 \leq \cdots \leq f_p$. In each iteration an attempt is made to improve the least favorable value of the function, f_1 in our case, by trying to find a new location $\boldsymbol{\beta}_1'$ for the vertex $\boldsymbol{\beta}_1$ such that $f_1 < f(\boldsymbol{\beta}_1')$.

Simplex Based Direct Search Iteration

First $\bar{\boldsymbol{\beta}}$ the centroid of the nonminimum vertices, $\boldsymbol{\beta}_k$, $k = 2, \ldots, p$, is obtained. The new location is then computed with one of the following operations along the direction $\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}_1$: reflection, expansion and contraction.

1. The *reflection* of $\boldsymbol{\beta}_1$, denoted $\boldsymbol{\beta}'$ (Figure 4.8a) is defined as

$$\boldsymbol{\beta}' = c_r \boldsymbol{\beta}_1 + (1 - c_r)\bar{\boldsymbol{\beta}} \tag{4.2.59}$$



(a) (b) (c) (d)

Figure 4.8. Basic operations in simplex based direct search. (a) Reflection. (b) Expansion. (c) Outside contraction. (d) Inside contraction.

where $c_r < 0$ is the reflection coefficient. If $f_2 < f(\boldsymbol{\beta}') \leq f_p$, then $\boldsymbol{\beta}'_1 = \boldsymbol{\beta}'$ and the next iteration is started.

2. If $f(\boldsymbol{\beta}') > f_p$, i.e., the reflection has produced a new maximum, the simplex is *expanded* by moving $\boldsymbol{\beta}'$ to $\boldsymbol{\beta}^*$ (Figure 4.8b)

$$\boldsymbol{\beta}^* = c_e \boldsymbol{\beta}' + (1 - c_e)\bar{\boldsymbol{\beta}} \tag{4.2.60}$$

where the expansion coefficient $c_e > 1$. If $f(\boldsymbol{\beta}^*) > f(\boldsymbol{\beta}')$ the expansion is successful and $\boldsymbol{\beta}'_1 = \boldsymbol{\beta}^*$. Else, $\boldsymbol{\beta}'_1 = \boldsymbol{\beta}'$. The next iteration is started.

3. If $f(\boldsymbol{\beta}') \leq f_2$, the vector $\boldsymbol{\beta}_{1n}$ is defined as either $\boldsymbol{\beta}_1$ or $\boldsymbol{\beta}'$, whichever has the larger associated function value, and a *contraction* is performed

$$\boldsymbol{\beta}^* = c_c \boldsymbol{\beta}_{1n} + (1 - c_c)\bar{\boldsymbol{\beta}} \; . \tag{4.2.61}$$

First, a contraction coefficient $0 < c_c < 1$ is chosen for outside contraction (Figure 4.8c). If $f(\boldsymbol{\beta}^*) > f(\boldsymbol{\beta}_{1n})$, then $\boldsymbol{\beta}'_1 = \boldsymbol{\beta}^*$ and the next iteration is started. Otherwise, an inside contraction is performed (Figure 4.8d) in which c_c is replaced with $-c_c$, and the condition $f(\boldsymbol{\beta}^*) > f(\boldsymbol{\beta}_{1n})$ is again verified.

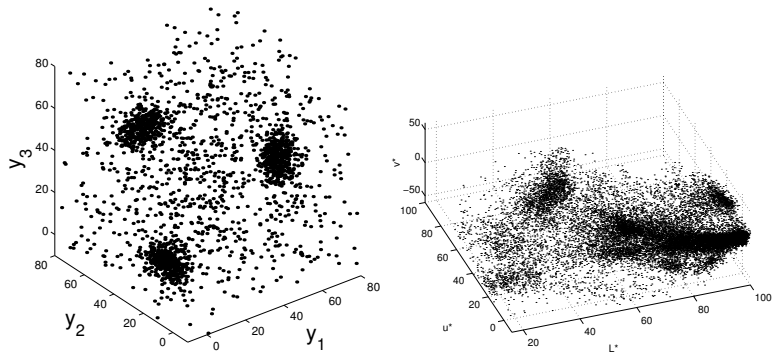
4. Should both contractions fail all the vertices are updated

$$\boldsymbol{\beta}_k \longleftarrow \frac{1}{2}\left(\boldsymbol{\beta}_k + \boldsymbol{\beta}_p\right) \qquad\qquad k = 1, \ldots, (p-1) \tag{4.2.62}$$

and the next iteration is started.

Recommended values for the coefficients are $c_r = -1,\ c_e = 1.5,\ c_c = 0.5$.

To assess the convergence of the search several stopping criteria can be employed. For example, the variance of the p function values f_k should fall below a threshold which is exponentially decreasing with the dimension of the space, or the ratio of the smallest and largest function values f_1/f_p should be close to one. Similarly, the volume of the simplex



(a) (b)

Figure 4.9. Multistructured data in the location estimation problem. (a) The "traditional" case. (b) A typical computer vision example.

should shrink below a dimension dependent threshold. In practice, the most effective stopping criteria are application specific incorporating additional information which was not used during the optimization.

In the previous sections we have analyzed the problem of robust estimation from a generic point of view. We can proceed now to examine the two classes of estimation problems: location and regression. In each case will introduce a new robust technique whose improved behavior was achieved by systematically exploiting the principles discussed so far.

4.3 Location Estimation

In this section will show that in the context of computer vision tasks often only nonparametric approaches can provide a robust solution for the location estimation problem. We employ a class of nonparametric techniques in which the data points are regarded as samples from an unknown probability density. The location estimates are then defined as the modes of this density. Explicit computation of the density is avoided by using the mean shift procedure.

4.3.1 Why Nonparametric Methods

The most general model of the location problem is that of multiple structures

$$\begin{array}{lll} k = 1, \ldots, K & m_1 = 1 \ \cdots \ m_{K+1} = n_1 + 1 & \\ \mathbf{y}_i^{(k)} = \mathbf{y}_{io}^{(k)} + \delta\mathbf{y}_i & \mathbf{y}_{io}^{(k)} = \boldsymbol{\theta}^{(k)} & i = m_k, \ldots, (m_{k+1} - 1) \\ \mathbf{y}_i & & i = (n_1 + 1), \ldots, n \end{array} \tag{4.3.1}$$

with no information being available about the nature of the inlier noise $\delta\mathbf{y}_i$, the $n - n_1$ outliers, or the number of structures present in the data K. The model (4.3.1) is also used

in *cluster analysis*, the equivalent pattern recognition problem. Clustering under its most general form is an unsupervised learning method of unknown categories from incomplete prior information [52, p. 242]. The books [52], [21, Chap. 10], [44, Sec.14.3] provide a complete coverage of the related pattern recognition literature.

Many of the pattern recognition methods are not adequate for data analysis in computer vision. To illustrate their limitations will compare the two data sets shown in Figure 4.9. The data in Figure 4.9a obeys what is assumed in traditional clustering methods when the proximity to a cluster center is measured as a function of Euclidean or Mahalanobis distances. In this case the shape of the clusters is restricted to elliptical and the inliers are assumed to be normally distributed around the true cluster centers. A different metric will impose a different shape on the clusters. The number of the structures (clusters) K, is a parameter to be supplied by the user and has a large influence on the quality of the results. While the value of K can be also derived from the data by optimizing a cluster validity index, this approach is not robust since it is based on (possibly erroneous) data partitions.

Expectation maximization (EM) is a frequently used technique today in computer vision to model the data. See [44, Sec.8.5.2] for a short description. The EM algorithm also relies on strong prior assumptions. A likelihood function, defined from a mixture of predefined (most often normal) probability densities, is maximized. The obtained partition of the data thus employs "tiles" of given shape. The number of required mixture components is often difficult to determine, and the association of these components with true cluster centers may not be obvious.

Examine now the data in Figure 4.9b in which the pixels of a color image were mapped into the three-dimensional $\mathrm{L^*u^*v^*}$ color space. The significant clusters correspond to similarly colored pixels in the image. The clusters have a large variety of shapes and their number is not obvious. Any technique which imposes a preset shape on the clusters will have difficulties to accurately separate K significant structures from the background clutter while simultaneously also having to determine the value of K.

Following our goal oriented approach toward robustness (Section 4.2.3) a location estimator should be declared robust only if it returns a satisfactory result. From the above discussion can be concluded that robustly solving location problems in computer vision often requires techniques which use the least possible amount of prior assumptions about the data. Such techniques belong to the family of *nonparametric* methods.

In nonparametric methods the n data points are regarded as outcomes from an (unknown) probability distribution $f(\mathbf{y})$. Each data point is assumed to have an equal probability

$$\mathrm{Prob}[\mathbf{y} = \mathbf{y}_i] = \frac{1}{n} \qquad i = 1, \ldots, n \; . \tag{4.3.2}$$

When several points have the same value, the probability is n^{-1} times the multiplicity. The ensemble of points defines the *empirical distribution* $f(\mathbf{y}|\mathbf{y}_1 \cdots \mathbf{y}_n)$ of the data. The empirical distribution is the nonparametric maximum likelihood estimate of the distribution from which the data was drawn [22, p.310]. It is also the "least committed" description of the data.

Every clustering technique exploits the fact that the clusters are the denser regions in the space. However, this observation can be pushed further in the class of nonparamet-

ric methods considered here, for which a region of higher density implies more probable outcomes of the random variable $\mathbf{y}$. Therefore, in each dense region the location estimate (cluster center) should be associated with the most probable value of $\mathbf{y}$, i.e., with the *local mode* of the empirical distribution

$$\hat{\boldsymbol{\theta}}^{(k)} = \underset{\mathbf{y}}{\operatorname{argmax}}_k \, f(\mathbf{y}|\mathbf{y}_1 \cdots \mathbf{y}_n) \qquad k = 1, \ldots, K \; . \tag{4.3.3}$$

Note that by detecting all the significant modes of the empirical distribution the number of clusters K is automatically determined. The mode based clustering techniques make extensive use of density estimation during data analysis.

4.3.2 Kernel Density Estimation

The modes of a random variable $\mathbf{y}$ are the local maxima of its probability density function $f(\mathbf{y})$. However, only the empirical distribution, the data points $\mathbf{y}_i$, $i = 1, \ldots, n$ are available. To accurately determine the locations of the modes, first a *continuous* estimate of the underlying density $\hat{f}(\mathbf{y})$ has to be defined. Later we will see that this step can be eliminated by directly estimating the gradient of the density (Section 4.3.3).

To estimate the probability density in $\mathbf{y}$ a small neighborhood is defined around $\mathbf{y}$. The neighborhood usually has a simple shape: cube, sphere or ellipsoid. Let its volume be V_y, and m_y be the number of data points inside. Then the density estimate is [21, Sec.4.2]

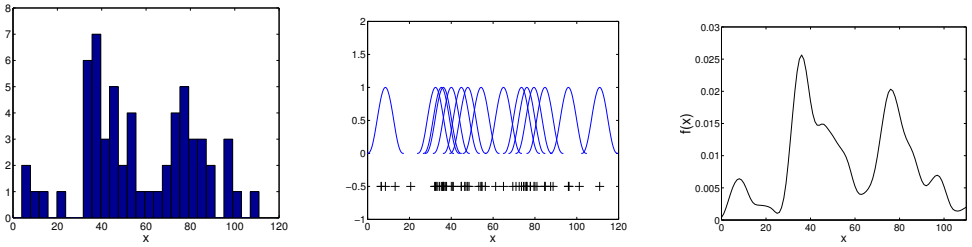
$$\hat{f}(\mathbf{y}) = \frac{m_y}{nV_y} \tag{4.3.4}$$

which can be employed in two different ways.

- In the *nearest neighbors* approach, the neigborhoods (the volumes V_y) are scaled to keep the number of points m_y constant. A mode corresponds to a location in which the neighborhood has the smallest volume.
- In the *kernel density* approach, the neigborhoods have the same volume V_y and the number of points m_y inside are counted. A mode corresponds to a location in which the neighborhood contains the largest number of points.

The *minimum volume ellipsoid* (MVE) robust location estimator proposed in statistics [90, p.258], is a technique related to the nearest neighbors approach. The ellipsoids are defined by elemental subsets obtained through random sampling, and the numerical optimization procedure discussed in Section 4.2.7 is employed. The location estimate is the center of the smallest ellipsoid which contains a given percentage of the data points. In a robust clustering method proposed in computer vision the MVE estimator was used to sequentially remove the clusters from the data, starting from the largest [53]. However, by imposing an elliptical shape for the clusters severe artifacts were introduced and the method was never successful in real vision applications.

For our goal of finding the local maxima of $\hat{f}(\mathbf{y})$, the kernel density methods are more suitable. Kernel density estimation is a widely used technique in statistics and pattern



(a) (b) (c)

Figure 4.10. Kernel density estimation. (a) Histogram of the data. (b) Some of the employed kernels. (c) The estimated density.

recognition, where it is also called the Parzen window method. See [93], [113] for a description in statistics, and [21, Sec.4.3] [44, Sec. 6.6] for a description in pattern recognition.

Will start with the simplest case of one-dimensional data. Let y_i, $i = 1, \ldots, n$, be scalar measurements drawn from an arbitrary probability distribution $f(y)$. The kernel density estimate $\hat{f}(y)$ of this distribution is obtained based on a *kernel function* $K(u)$ and a *bandwidth* h as the average

$$\hat{f}(y) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{y - y_i}{h}\right) . \tag{4.3.5}$$

Only the class of symmetric kernel functions with bounded support will be considered. They satisfy the following properties

$$\begin{aligned} K(u) &= 0 \quad \text{for } |u| > 1 \qquad & \int_{-1}^{1} K(u) = 1 \\ K(u) &= K(-u) \geq 0 \qquad & K(u_1) \geq K(u_2) \quad \text{for } |u_1| \leq |u_2| . \end{aligned} \tag{4.3.6}$$

Other conditions on the kernel function or on the density to be estimated [113, p.18], are of less significance in practice. The even symmetry of the kernel function allows us to define its *profile* $k(u)$

$$K(u) = c_k k(u^2) \qquad k(u) \geq 0 \text{ for } 0 \leq u \leq 1 \tag{4.3.7}$$

where c_k is a normalization constant determined by (4.3.6). The shape of the kernel implies that the profile is a monotonically decreasing function.

The kernel density estimate is a *continuous* function derived from the discrete data, the empirical distribution. An example is shown in Figure 4.10. When instead of the histogram of the n points (Figure 4.10a) the data is represented as an ordered list (Figure 4.10b, bottom), we are in fact using the empirical distribution. By placing a kernel in each point (Figure 4.10b) the data is convolved with the symmetric kernel function. The density estimate in a given location is the average of the contributions from each kernel (Figure

4.10c). Since the employed kernel has a finite support, not all the points contribute to a density estimate. The bandwidth h scales the size of the kernels, i.e., the number of points whose contribution is averaged when computing the estimate. The bandwidth thus controls the amount of smoothing present in $\hat{f}(y)$.

For multivariate measurements $\mathbf{y}_i \in R^p$, in the most general case, the bandwidth h is replaced by a symmetric, positive definite bandwidth matrix H. The estimate of the probability distribution at location $\mathbf{y}$ is still computed as the average

$$\hat{f}(\mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n} K_{\mathrm{H}}\left(\mathbf{y}-\mathbf{y}_i\right) \tag{4.3.8}$$

where the bandwidth matrix H scales the kernel support to be radial symmetric, i.e., to have the desired elliptical shape and size

$$K_{\mathrm{H}}(\mathbf{u}) = [\det[\mathrm{H}]]^{-1/2} K(\mathrm{H}^{-1/2}\mathbf{u}) \; . \tag{4.3.9}$$

Since only circular symmetric prototype kernels $K(\mathbf{u})$ will be considered, we have, using the profile $k(u)$

$$K(\mathbf{u}) = c_{k,p} k(\mathbf{u}^{\top}\mathbf{u}) \; . \tag{4.3.10}$$

From (4.3.8), taking into account (4.3.10) and (4.3.9) results

$$\begin{aligned} \hat{f}(\mathbf{y}) &= \frac{c_{k,p}}{n[\det[\mathrm{H}]]^{1/2}} \sum_{i=1}^{n} k\left((\mathbf{y}-\mathbf{y}_i)^{\top}\mathrm{H}^{-1}(\mathbf{y}-\mathbf{y}_i)\right) \\ &= \frac{c_{k,p}}{n[\det[\mathrm{H}]]^{1/2}} \sum_{i=1}^{n} k\left(\mathrm{d}[\mathbf{y},\mathbf{y}_i,\mathrm{H}]^2\right) \end{aligned} \tag{4.3.11}$$

where the expression $\mathrm{d}[\mathbf{y},\mathbf{y}_i,\mathrm{H}]^2$ denotes the squared Mahalanobis distance from $\mathbf{y}$ to $\mathbf{y}_i$. The case $\mathrm{H} = h^2\mathrm{I}_p$ is the most often used. The kernels then have a circular support whose radius is controlled by the bandwidth h and (4.3.8) becomes

$$\hat{f}_K(\mathbf{y}) = \frac{1}{nh^p}\sum_{i=1}^{n} K\left(\frac{\mathbf{y}-\mathbf{y}_i}{h}\right) = \frac{c_{k,p}}{nh^p}\sum_{i=1}^{n} k\left(\left\|\frac{\mathbf{y}-\mathbf{y}_i}{h}\right\|^2\right) \tag{4.3.12}$$

where the dependence of the density estimate on the kernel was made explicit.

The quality of a density estimate $\hat{f}(\mathbf{y})$ is assessed in statistics using the *asymptotic mean integrated error* (AMISE), i.e., the integrated mean square error

$$\mathrm{MISE}(\mathbf{y}) = \int \mathrm{E}\left[f(\mathbf{y}) - \hat{f}(\mathbf{y})\right]^2 d\mathbf{y} \; . \tag{4.3.13}$$

between the true density and its estimate for $n \to \infty$ while $h \to 0$ at a slower rate. The expectation is taken over all data sets of size n. Since the bandwidth h of a circular symmetric kernel has a strong influence on the quality of $\hat{f}(\mathbf{y})$, the bandwidth minimizing

an approximation of the AMISE error is of interest. Unfortunately, this bandwidth depends on $f(\mathbf{y})$, the unknown density [113, Sec.4.3].

For the univariate case several practical rules are available [113, Sec.3.2]. For example, the information about $f(y)$ is substituted with $\hat{\sigma}$, a robust scale estimate derived from the data

$$\hat{h} = \left[\frac{243R(K)}{35\mu_2(K)^2 n}\right]^{1/5} \hat{\sigma} \tag{4.3.14}$$

where

$$\mu_2(K) = \int_{-1}^{1} u^2 K(u)du \qquad R(K) = \int_{-1}^{1} K(u)^2 du \tag{4.3.15}$$

The scale estimate $\hat{\sigma}$ will be discussed in Section 4.4.3.

For a given bandwidth the AMISE measure is minimized by the *Epanechnikov* kernel [113, p.104] having the profile

$$k_E(u) = \begin{cases} 1-u & 0 \le u \le 1 \\ 0 & u > 1 \end{cases} \tag{4.3.16}$$

which yields the kernel

$$K_E(\mathbf{y}) = \begin{cases} \frac{1}{2}c_p^{-1}(p+2)(1-\|\mathbf{y}\|^2) & \|\mathbf{y}\| \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.3.17}$$

where c_p is the volume of the p-dimensional unit sphere. Other kernels can also be defined. The *truncated normal* has the profile

$$k_N(u) = \begin{cases} e^{-au} & 0 \le u \le 1 \\ 0 & u > 1 \end{cases} \tag{4.3.18}$$

where a is chosen such that e^{-a} is already negligible small. Neither of the two profiles defined above have continuous derivatives at the boundary $u = 1$. This condition is satisfied (for the first two derivatives) by the *biweight* kernel having the profile

$$k_B(u) = \begin{cases} (1-u)^3 & 0 \le u \le 1 \\ 0 & u > 1 \end{cases} \tag{4.3.19}$$

Its name here is taken from robust statistics, in the kernel density estimation literature it is called the triweight kernel [113, p.31].

The bandwidth matrix H is the critical parameter of a kernel density estimator. For example, if the region of summation (bandwidth) is too large, significant features of the distribution, like multimodality, can be missed by oversmoothing. Furthermore, locally the data can have very different densities and using a single bandwidth matrix often is not enough to obtain a satisfactory estimate.

There are two ways to adapt the bandwidth to the local structure, in each case the adaptive behavior being achieved by first performing a pilot density estimation. The bandwidth

matrix can be either associated with the location $\mathbf{y}$ in which the distribution is to be estimated, or each measurement $\mathbf{y}_i$ can be taken into account in (4.3.8) with its own bandwidth matrix

$$\hat{f}_K(\mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n} K_{\mathrm{H}_i}(\mathbf{y}-\mathbf{y}_i) \ . \tag{4.3.20}$$

It can be shown that (4.3.20), called the *sample point* density estimator, has superior statistical properties [39].

The local maxima of the density $f(\mathbf{y})$ are by definition the roots of the equation

$$\nabla f(\mathbf{y}) = 0 \tag{4.3.21}$$

i.e., the zeros of the density gradient. Note that the converse is not true since any *stationary point* of $f(\mathbf{y})$ satisfies (4.3.21). The true density, however, is not available and in practice the estimate of the gradient $\hat{\nabla} f(\mathbf{y})$ has to be used.

In the next section we describe the *mean shift* technique which avoids explicit computation of the density estimate when solving (4.3.21). The mean shift procedure also associates each data point to the nearest density maximum and thus performs a nonparametric clustering in which the shape of the clusters is not set a priori.

4.3.3 Adaptive Mean Shift

The mean shift method was described in several publications [16], [18], [17]. Here we are considering its most general form in which each measurement $\mathbf{y}_i$ be associated with a known bandwidth matrix H_i, $i = 1,\ldots,n$. Taking the gradient of the sample point density estimator (4.3.20) we obtain, after recalling (4.3.11) and exploiting the linearity of the expression

$$\begin{aligned} \hat{\nabla} f_K(\mathbf{y}) &\equiv \nabla \hat{f}_K(\mathbf{y}) \\ &= \frac{2c_{k,p}}{n}\sum_{i=1}^{n}[\det[\mathrm{H}_i]]^{-1/2}\,\mathrm{H}_i^{-1}(\mathbf{y}-\mathbf{y}_i)\,k'\left(\mathrm{d}[\mathbf{y},\mathbf{y}_i,\mathrm{H}_i]^2\right) \ . \end{aligned} \tag{4.3.22}$$

The function $g(x) = -k'(x)$ satisfies the properties of a profile, and thus we can define the kernel $G(\mathbf{u}) = c_{g,p}\,g(\mathbf{u}^\top\mathbf{u})$. For example, for the Epanechnikov kernel the corresponding new profile is

$$g_E(u) = \begin{cases} 1 & 0 \le u \le 1 \\ 0 & u > 1 \end{cases} \tag{4.3.23}$$

and thus $G_E(\mathbf{u})$ is the uniform kernel. For convenience will introduce the notation

$$\mathrm{Q}_i(\mathbf{y}) = \det[\mathrm{H}_i]^{-1/2}\,\mathrm{H}_i^{-1} g\left(\mathrm{d}[\mathbf{y},\mathbf{y}_i,\mathrm{H}_i]^2\right) \ . \tag{4.3.24}$$

From the definition of $g(u)$ and (4.3.7)

$$\mathrm{Q}_i(\mathbf{y}) = \mathrm{O} \qquad \mathrm{d}[\mathbf{y},\mathbf{y}_i,\mathrm{H}_i] > 1 \ . \tag{4.3.25}$$

Then (4.3.22) can be written as

$$\hat{\nabla} f_K(\mathbf{y}) = \frac{2c_{k,p}}{n} \left(\sum_{i=1}^{n} \mathrm{Q}_i(\mathbf{y}) \right) \left[\left(\sum_{i=1}^{n} \mathrm{Q}_i(\mathbf{y}) \right)^{-1} \sum_{i=1}^{n} \mathrm{Q}_i(\mathbf{y}) \mathbf{y}_i - \mathbf{y} \right] \tag{4.3.26}$$

and the roots of the equation (4.3.21) are the solutions of

$$\mathbf{y} = \left(\sum_{i=1}^{n} \mathrm{Q}_i(\mathbf{y}) \right)^{-1} \sum_{i=1}^{n} \mathrm{Q}_i(\mathbf{y}) \mathbf{y}_i \tag{4.3.27}$$

which can be solved only iteratively

$$\mathbf{y}^{[l+1]} = \left(\sum_{i=1}^{n} \mathrm{Q}_i\left(\mathbf{y}^{[l]}\right) \right)^{-1} \sum_{i=1}^{n} \mathrm{Q}_i\left(\mathbf{y}^{[l]}\right) \mathbf{y}_i \qquad l = 0, 1, \ldots \tag{4.3.28}$$

The meaning of an iteration becomes apparent if we consider the particular case $\mathrm{H}_i = h_i^2 \mathrm{I}$ yielding

$$\mathbf{y} = \frac{\sum_{i=1}^{n} \mathbf{y}_i \, h_i^{-(p+2)} \, g\left(\left\| \frac{\mathbf{y} - \mathbf{y}_i}{h_i} \right\|^2 \right)}{\sum_{i=1}^{n} h_i^{-(p+2)} \, g\left(\left\| \frac{\mathbf{y} - \mathbf{y}_i}{h_i} \right\|^2 \right)} \tag{4.3.29}$$

and which becomes when all $h_i = h$

$$\mathbf{y} = \frac{\sum_{i=1}^{n} \mathbf{y}_i \, g\left(\left\| \frac{\mathbf{y} - \mathbf{y}_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{y} - \mathbf{y}_i}{h} \right\|^2 \right)} \, . \tag{4.3.30}$$

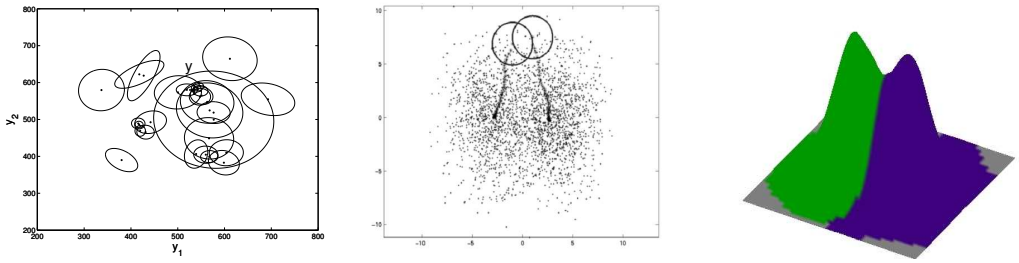
From (4.3.25) we see that at every step only a *local* weighted mean is computed. The robustness of the mode detection method is the direct consequence of this property. In the next iteration the computation is repeated centered on the previously computed mean. The difference between the current and previous locations, the vector

$$\mathbf{m}_G^{[l+1]} = \mathbf{y}^{[l+1]} - \mathbf{y}^{[l]} \qquad l = 0, 1, \ldots \tag{4.3.31}$$

is called the *mean shift* vector, where the fact that the weighted averages are computed with the kernel G was made explicit. Adapting (4.3.26) to the two particular cases above it can be shown that

$$\mathbf{m}_G^{[l+1]} = c \frac{\hat{\nabla} f_K(\mathbf{y}^{[l]})}{\hat{f}_G(\mathbf{y}^{[l]})} \tag{4.3.32}$$

where c is a positive constant. Thus, the mean shift vector is aligned with the gradient estimate of the density, and the window of computations is always moved toward regions



(a) (b) (c)

Figure 4.11. The main steps in mean shift based clustering. (a) Computation of the weighted mean in the general case. (b) Mean shift trajectories of two points in a bimodal data. (c) Basins of attraction.

of higher density. See [17] for the details. A relation similar to (4.3.32) still holds in the general case, but now the mean shift and gradient vectors are connected by a linear transformation.

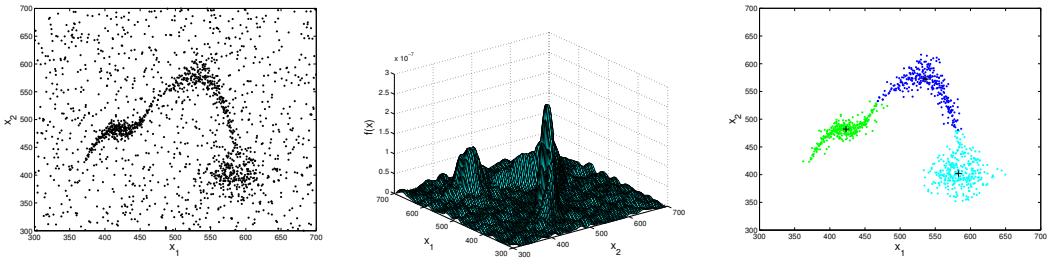
In the mean shift procedure the user controls the resolution of the data analysis by providing the bandwidth information. Since most often circular symmetric kernels are used, only the bandwith parameters h_i are needed.

Mean Shift Procedure

1. Choose a data point $\mathbf{y}_i$ as the initial $\mathbf{y}^{[0]}$.
2. Compute $\mathbf{y}^{[l+1]}$, $l = 0, 1, \ldots$ the weighted mean of the points at less than unit Mahalanobis distance from $\mathbf{y}^{[l]}$. Each point is considered with its own metric.
3. Verify if $\left\| \mathbf{m}_G^{[l+1]} \right\|$ is less than the tolerance. If yes, stop.
4. Replace $\mathbf{y}^{[l]}$ with $\mathbf{y}^{[l+1]}$, i.e., move the processing toward a region with higher point density. Return to Step 2.

The most important properties of the mean shift procedure are illustrated graphically in Figure 4.11. In Figure 4.11a the setup of the weighted mean computation in the general case is shown. The kernel associated with a data point is nonzero only within the elliptical region centered on that point. Thus, only those points contribute to the weighted mean in $\mathbf{y}$ whose kernel support contains $\mathbf{y}$.

The evolution of the iterative procedure is shown in Figure 4.11b for the simplest case of identical circular kernels (4.3.30). When the locations of the points in a window are averaged, the result is biased toward the region of higher point density in that window. By moving the window into the new position we move uphill on the density surface. The



(a) (b) (c)

Figure 4.12. An example of clustering using the mean shift procedure. (a) The two-dimensional input. (b) Kernel density estimate of the underlying distribution. (c) The basins of attraction of the three significant modes (marked '+').

mean shift procedure is a gradient ascent type technique. The processing climbs toward the highest point on the side of the density surface on which the initial position $\mathbf{y}^{[0]}$ was placed. At convergence (which can be proven) the local maximum of the density, the sought mode, is detected.

The two initializations in Figure 4.11b are on different components of this mixture of two Gaussians. Therefore, while the two mean shift procedures start from nearby locations, they converge to different modes, both of which are accurate location estimates.

A nonparametric classification of the data into clusters can be obtained by starting a mean shift procedure from every data point. A set of points converging to nearby locations defines the *basin of attraction* of a mode. Since the points are processed independently the shape of the basin of attraction is not restricted in any way. The basins of attraction of the two modes of a Gaussian mixture (Figure 4.11c) were obtained without using the nature of the distributions.

The two-dimensional data in Figure 4.12a illustrates the power of the mean shift based clustering. The three clusters have arbitrary shapes and the background is heavily cluttered with outliers. Traditional clustering methods would have difficulty yielding satisfactory results. The three significant modes in the data are clearly revealed in a kernel density estimate (Figure 4.12b). The mean shift procedure detects all three modes, and the associated basins of attraction provide a good delineation of the individual clusters (Figure 4.12c). In practice, using only a subset of the data points suffices for an accurate delineation. See [16] for details of the mean shift based clustering.

The original mean shift procedure was proposed in 1975 by Fukunaga and Hostetler [32]. See also [31, p.535]. It came again into attention with the paper [10]. In spite of its excellent qualities, mean shift is less known in the statistical literature. The book [93, Sec.6.2.2] discusses [32], and a similar technique is proposed in [11] for bias reduction in density estimation.

The simplest, *fixed bandwith* mean shift procedure in which all $\mathrm{H}_i = h^2\mathrm{I}_p$, is the one most frequently used in computer vision applications. The adaptive mean shift procedure discussed in this section, however, is not difficult to implement with circular symmetric

kernels, i.e., $\mathrm{H}_i = h_i^2 \mathrm{I}_p$. The bandwidth value h_i associated with the data point $\mathbf{y}_i$ can be defined as the distance to the k-th neighbor, i.e., for the pilot density estimation the nearest neighbors approach is used. An implementation for high dimensional spaces is described in [35]. Other, more sophisticated methods for local bandwidth selection are described in [15], [18]. Given the complexity of the visual data, such methods, which are based on assumptions about the local structure, may not provide any significant gain in performance.

4.3.4 Applications

We will sketch now two applications of the fixed bandwith mean shift procedure, i.e., circular kernels with $\mathrm{H}_i = h^2 \mathrm{I}_p$.

- discontinuity preserving filtering and segmentation of color images;
- tracking of nonrigid objects in a color image sequence.

These applications are the subject of [17] and [19] respectively, which should be consulted for details.

An image can be regarded as a vector field defined on the two-dimensional lattice. The dimension of the field is one in the gray level case and three for color images. The image coordinates belong to the *spatial* domain, while the gray level or color information is in the *range* domain. To be able to use in the mean shift procedure circular symmetric kernels the validity of an Euclidean metric must be verified for both domains. This is most often true in the spatial domain and for gray level images in the range domain. For color images, mapping the RGB input into the $\mathrm{L^*u^*v^*}$ (or $\mathrm{L^*a^*b^*}$) color space provides the closest possible Euclidean approximation for the perception of color differences by human observers.

The goal in image filtering and segmentation is to generate an accurate piecewise constant representation of the input. The constant parts should correspond in the input image to contiguous regions with similarly colored pixels, while the discontinuities to significant changes in color. This is achieved by considering the spatial and range domains jointly. In the joint domain the basin of attraction of a mode corresponds to a contiguous homogeneous region in the input image and the valley between two modes most often represents a significant color discontinuity in the input. The joint mean shift procedure uses a product kernel

$$K(\mathbf{y}) = \frac{c}{h_s^2 h_r^q} k\left(\left\|\frac{\mathbf{y}^s}{h_s}\right\|^2\right) k\left(\left\|\frac{\mathbf{y}^r}{h_r}\right\|^2\right) \tag{4.3.33}$$

where $\mathbf{y}^s$ and $\mathbf{y}^r$ are the spatial and the range parts of the feature vector, $k(u)$ is the profile of the kernel used in both domains (though they can also differ), h_s and h_r are the employed bandwidths parameters, and c is the normalization constant. The dimension of the range domain q, is one for the gray level and three for the color images. The user sets the value of the two bandwidth parameters according to the desired resolution of the image analysis.

In *discontinuity preserving filtering* every pixel is allocated to the nearest mode in the joint domain. All the pixels in the basin of attraction of the mode get the range value of that mode. From the spatial arrangement of the basins of attraction the region adjacency graph (RAG) of the input image is then derived. A transitive closure algorithm is performed on

the RAG and the basins of attraction of adjacent modes with similar range values are fused. The result is the *segmented* image.

The gray level image example in Figure 4.13 illustrates the role of the mean shift procedure. The small region of interest (ROI) in Figure 4.13a is shown in a wireframe representation in Figure 4.13b. The three-dimensional G kernel used in the mean shift procedure (4.3.30) is in the top-left corner. The kernel is the product of two uniform kernels: a circular symmetric two-dimensional kernel in the spatial domain and a one-dimensional kernel for the gray values.

At every step of the mean shift procedure, the average of the 3D data points is computed and the kernel is moved to the next location. When the kernel is defined at a pixel on the high plateau on the right in Figure 4.13b, adjacent pixels (neighbors in the spatial domain) have very different gray level values and will *not* contribute to the average. This is how the mean shift procedure achieves the discontinuity preserving filtering. Note that the probability density function whose local mode is sought cannot be visualized since it would require a four-dimensional space, the fourth dimension being that of the density.

The result of the segmentation for the ROI is shown in Figure 4.13c, and for the entire image in Figure 4.13d. A more accurate segmentation is obtained if edge information is incorporated into the mean shift procedure (Figures 4.13e and 4.13f). The technique is described in [13].

A color image example is shown in Figure 4.14. The input has large homogeneous regions, and after filtering (Figures 4.14b and 4.14c) many of the delineated regions already correspond to semantically meaningful parts of the image. However, this is more the exception than the rule in filtering. A more realistic filtering process can be observed around the windows, where many small regions (basins of attraction containing only a few pixels) are present. These regions are either fused or attached to a larger neighbor during the transitive closure process on the RAG, and the segmented image (Figures 4.14d and 4.14e) is less cluttered. The quality of any segmentation, however, can be assessed only through the performance of subsequent processing modules for which it serves as input.

The discontinuity preserving filtering and the image segmentation algorithm were integrated together with a novel edge detection technique [74] in the *Edge Detection and Image SegmentatiON* (EDISON) system [13]. The C++ source code of EDISON is available on the web at

`www.caip.rutgers.edu/riul/`

The second application of the mean shift procedure is *tracking* of a dynamically changing neighborhood in a sequence of color images. This is a critical module in many object recognition and surveillance tasks. The problem is solved by analyzing the image sequence as pairs of two consecutive frames. See [19] for a complete discussion.

The neighborhood to be tracked, i.e., the *target model* in the first image contains n_a pixels. We are interested only in the amount of relative translation of the target between the two frames. Therefore, without loss of generality the target model can be considered centered on $\mathbf{y}_a = \mathbf{0}$. In the next frame, the *target candidate* is centered on $\mathbf{y}_b$ and contains n_b pixels.

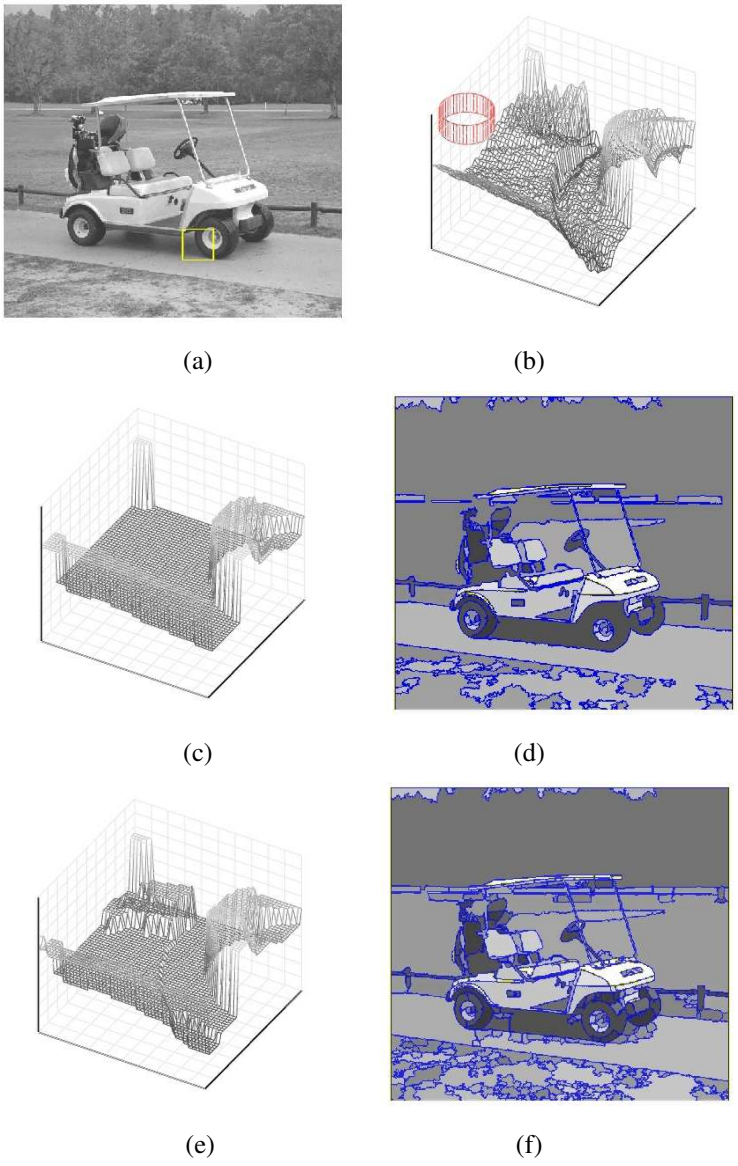


Figure 4.13. The image segmentation algorithm. (a) The gray level input image with a region of interest (ROI) marked. (b) The wireframe representation of the ROI and the 3D window used in the mean shift procedure. (c) The segmented ROI. (d) The segmented image. (e) The segmented ROI when local discontinuity information is integrated into the mean shift procedure. (f) The segmented image.

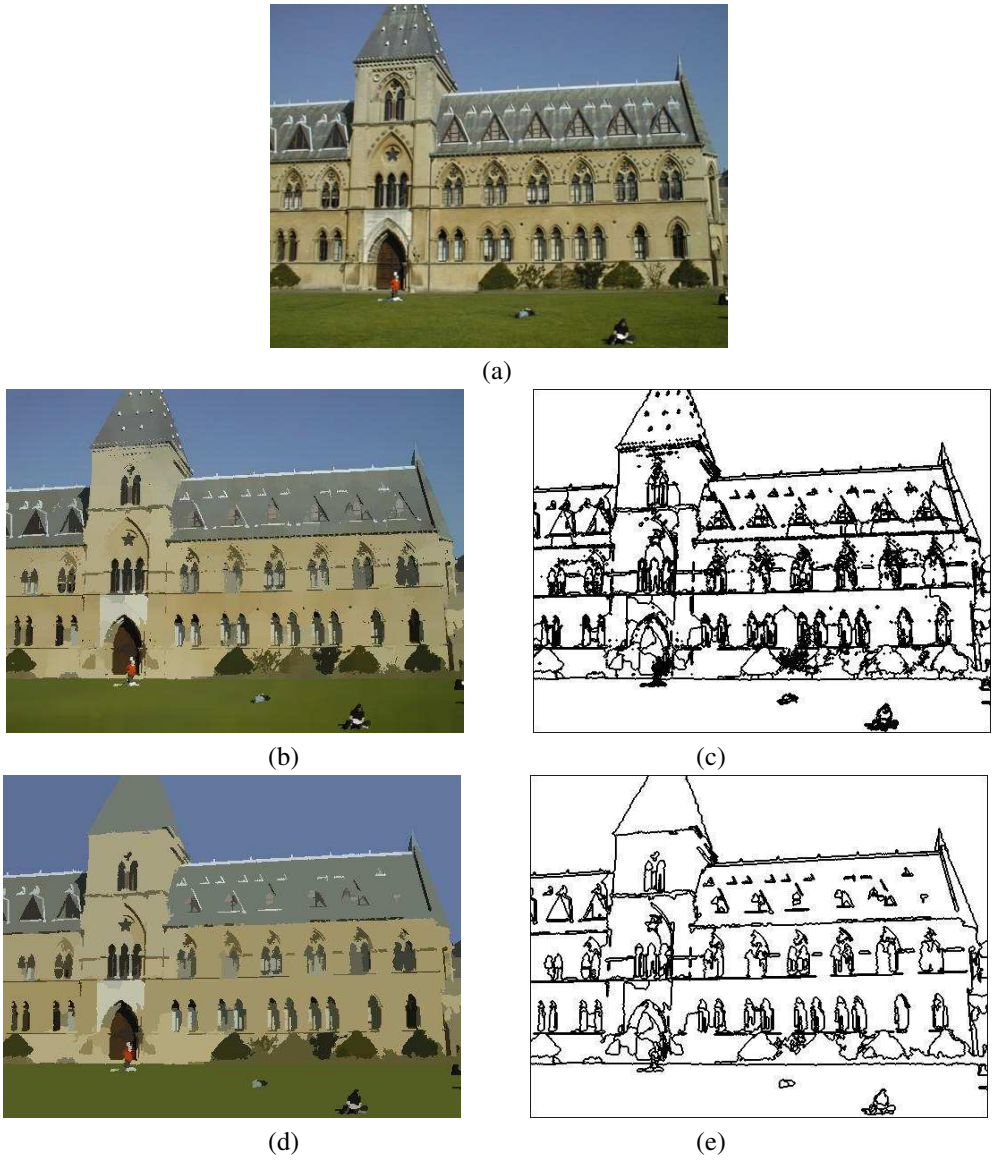


Figure 4.14. A color image filtering/segmentation example. (a) The input image. (b) The filtered image. (c) The boundaries of the delineated regions. (d) The segmented image. (e) The boundaries of the delineated regions.

In both color images kernel density estimates are computed in the joint five-dimensional domain. In the spatial domain the estimates are defined in the center of the neighborhoods, while in the color domain the density is sampled at m locations $\mathbf{c}$. Let $c = 1, \ldots, m$ be a scalar hashing index of these three-dimensional sample points. A kernel with profile $k(u)$ and bandwidths h_a and h_b is used in the spatial domain. The sampling in the color domain is performed with the Kronecker delta function $\delta(u)$ as kernel.

The result of the two kernel density estimations are the two discrete color densities associated with the target in the two images. For $c = 1, \ldots, m$

$$\text{model:} \qquad \hat{f}_a(c) = A \sum_{i=1}^{n_a} k\left(\left\| \frac{\mathbf{y}_{a,i}}{h_a} \right\|^2\right) \delta\left[\mathbf{c}(\mathbf{y}_{a,i}) - \mathbf{c}\right] \tag{4.3.34}$$

$$\text{candidate:} \qquad \hat{f}_b(c, \mathbf{y}_b) = B \sum_{i=1}^{n_b} k\left(\left\| \frac{\mathbf{y}_b - \mathbf{y}_{b,i}}{h_b} \right\|^2\right) \delta\left[\mathbf{c}(\mathbf{y}_{b,i}) - \mathbf{c}\right] \tag{4.3.35}$$

where, $\mathbf{c}(\mathbf{y})$ is the color vector of the pixel at $\mathbf{y}$. The normalization constants A, B are determined such that

$$\sum_{c=1}^{m} \hat{f}_a(c) = 1 \qquad\qquad \sum_{c=1}^{m} \hat{f}_b(c, \mathbf{y}_b) = 1 \; . \tag{4.3.36}$$

The normalization assures that the template matching score between these two discrete signals is

$$\rho(\mathbf{y}_b) = \sum_{c=1}^{m} \sqrt{\hat{f}_a(c)\, \hat{f}_b(c, \mathbf{y}_b)} \tag{4.3.37}$$

and it can be shown that

$$d(\mathbf{y}_b) = \sqrt{1 - \rho(\mathbf{y}_b)} \tag{4.3.38}$$

is a metric distance between $\hat{f}_a(c)$ and $\hat{f}_b(c, \mathbf{y}_b)$

To find the location of the target in the second image, the distance (4.3.38) has to be minimized over $\mathbf{y}_b$, or equivalently (4.3.37) has to be maximized. That is, the local maximum of $\rho(\mathbf{y}_b)$ has to be found by performing a search in the second image. This search is implemented using the mean shift procedure.

The local maximum is a root of the template matching score gradient

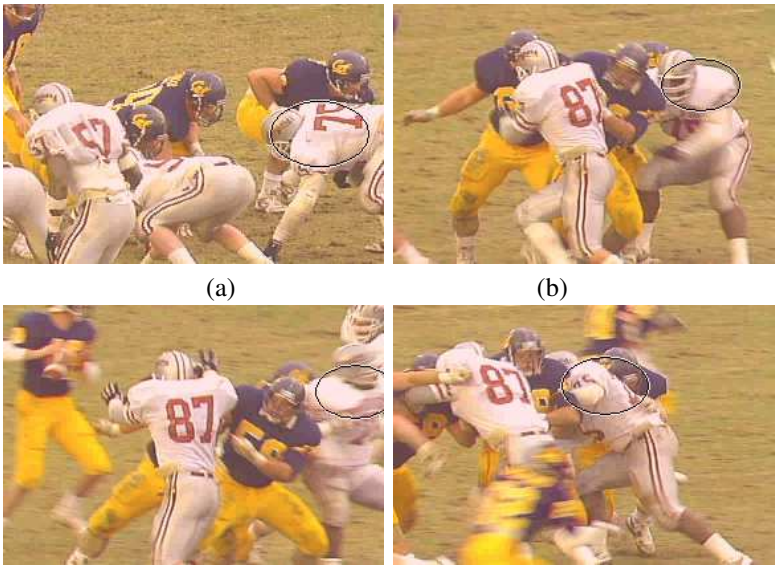
$$\nabla \rho(\mathbf{y}_b) = \frac{1}{2} \sum_{c=1}^{m} \nabla \hat{f}_b(c, \mathbf{y}_b) \sqrt{\frac{\hat{f}_a(c)}{\hat{f}_b(c, \mathbf{y}_b)}} = 0 \; . \tag{4.3.39}$$

Taking into account (4.3.35) yields

$$\sum_{c=1}^{m} \sum_{i=1}^{n_b} (\mathbf{y}_b - \mathbf{y}_{b,i})\, k'\left(\left\| \frac{\mathbf{y}_b - \mathbf{y}_{b,i}}{h} \right\|^2\right) \delta\left[\mathbf{c}(\mathbf{y}_{b,i}) - \mathbf{c}\right] \sqrt{\frac{\hat{f}_a(c)}{\hat{f}_b(c, \mathbf{y}_b)}} = 0 \; . \tag{4.3.40}$$

As in Section 4.3.3 we can introduce the profile $g(u) = -k'(u)$ and define the weights

$$q_i(\mathbf{y}_b) = \sum_{c=1}^{m} \sqrt{\frac{\hat{f}_a(c)}{\hat{f}_b(c, \mathbf{y}_b)}}\, \delta\left[\mathbf{c}(\mathbf{y}_{b,i}) - \mathbf{c}\right] \tag{4.3.41}$$



(a) (b)

(c) (d)

Figure 4.15. An example of the tracking algorithm. (a) The first frame of a color image sequence with the target model manually defined as the marked elliptical region. (b) to (d) Localization of the target in different frames.

and obtain the iterative solution of (4.3.39) from

$$\mathbf{y}_b^{[l+1]} = \frac{\sum_{i=1}^{n_b} \mathbf{y}_{b,i}^{[l]} \; q_i\left(\mathbf{y}_b^{[l]}\right) \; g\left(\left\| \frac{\mathbf{y}_b^{[l]} - \mathbf{y}_{b,i}}{h_b} \right\|^2\right)}{\sum_{i=1}^{n_b} q_i\left(\mathbf{y}_b^{[l]}\right) \; g\left(\left\| \frac{\mathbf{y}_b^{[l]} - \mathbf{y}_{b,i}}{h_b} \right\|^2\right)} \tag{4.3.42}$$

which is a mean shift procedure, the only difference being that at each step the weights (4.3.41) are also computed.

In Figure 4.15 four frames of an image sequence are shown. The target model, defined in the first frame (Figure 4.15a), is successfully tracked throughout the sequence. As can be seen, the localization is satisfactory in spite of the target candidates' color distribution being significantly different from that of the model. While the model can be updated as we move along the sequence, the main reason for the good performance is the small amount of translation of the target region between two consecutive frames. The search in the second image always starts from the location of the target model center in the first image. The mean shift procedure then finds the *nearest* mode of the template matching score, and with high probability this is the target candidate location we are looking for. See [19] for more examples and extensions of the tracking algorithm, and [14] for a version with automatic

bandwidth selection.

The robust solution of the location estimation problem presented in this section put the emphasis on employing the least possible amount of a priori assumptions about the data and belongs to the class of nonparametric techniques. Nonparametric techniques require a larger number of data points supporting the estimation process than their parametric counterparts. In parametric methods the data is more constrained, and as long as the model is obeyed the parametric methods are better in extrapolating over regions where data is not available. However, if the model is not correct a parametric method will still impose it at the price of severe estimation errors. This important trade-off must be kept in mind when feature space analysis is used in a complex computer vision task.

4.4 Robust Regression

The linear errors-in-variables (EIV) regression model (Section 4.2.6) is employed for the discussion of the different regression techniques. In this model the inliers are measured as

$$\mathbf{y}_i = \mathbf{y}_{io} + \delta\mathbf{y}_i \quad \mathbf{y}_i \in \mathcal{R}^p \quad \delta\mathbf{y}_i \sim GI(\mathbf{0}, \sigma^2 \mathrm{I}_p) \qquad i = 1, \ldots, n_1 \tag{4.4.1}$$

and their true values obey the constraints

$$g(\mathbf{y}_{io}) = \mathbf{y}_{io}^\top \boldsymbol{\theta} - \alpha = 0 \quad i = 1, \ldots, n_1 \qquad \|\boldsymbol{\theta}\| = 1 \qquad \alpha \geq 0\,. \tag{4.4.2}$$

The number of inliers must be much larger than the number of free parameters of the model, $n_1 \gg p$. Nothing is assumed about the $n - n_1$ outliers.

After a robust method selects the inliers they are often postprocessed with a nonrobust technique from the least squares (LS) family to obtain the final parameter. Therefore, we start by discussing the LS estimators. Next, the family of M-estimators is introduced and the importance of the scale parameter related to the noise of the inliers is emphasized.

All the robust regression methods popular today in computer vision can be described within the framework of M-estimation and thus their performance also depends on the accuracy of the scale parameter. To avoid this deficiency, we approach M-estimation in a different way and introduce the pbM-estimator which does not require the user to provide the value of the scale.

In Section 4.2.5 it was shown that when a nonlinear EIV regression model is processed as a linear model in the carriers, the associated noise is heteroscedastic. Since the robust methods discussed in this section assume the model (4.4.1) and (4.4.2), they return biased estimates if employed for solving nonlinear EIV regression problems. However, this does not mean they should not be used! The role of any robust estimator is only to establish a satisfactory inlier/outlier dichotomy. As long as most of the inliers were recovered from the data, postprocessing with the proper nonlinear (and nonrobust) method will provide the correct estimates.

Regression in the presence of multiple structures in the data will not be considered beyond the particular case of two structures in the context of structured outliers. We will show why all the robust regression methods fail to handle such data once the measurement noise becomes large.

Each of the regression techniques in this section is related to one of the objective functions described in Section 4.2.2. Using the same objective function location models can also be estimated, but we will not discuss these location estimators. For example, many of the traditional clustering methods belong to the least squares family [52, Sec.3.3.2], or there is a close connection between the mean shift procedure and M-estimators of location [17].

4.4.1 Least Squares Family

We have seen in Section 4.2.3 that the least squares family of estimators is not robust since its objective function $\mathcal{J}_{LS}$ (4.2.22) is a symmetric function in *all* the measurements. Therefore, in this section will assume that the data contains only inliers, i.e., $n = n_1$.

The parameter estimates of the linear EIV regression model are obtained by solving the minimization

$$[\,\hat{\alpha},\,\hat{\boldsymbol{\theta}}\,] = \underset{\alpha,\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{y}_i - \mathbf{y}_{io}\|^2 = \underset{\alpha,\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} g(\mathbf{y}_i)^2 \tag{4.4.3}$$

subject to (4.4.2). The minimization yields the *total least squares* (TLS) estimator. For an in-depth analysis of the TLS estimation see the book [112]. Related problems were already discussed in the nineteenth century [33, p.30], though the method most frequently used today, based on the singular value decomposition (SVD) was proposed only in 1970 by Golub and Reinsch [37]. See the book [38] for the linear algebra background.

To solve the minimization problem (4.4.3) will define the $n \times p$ matrices of the measurements Y and of the true values Y_o

$$\mathrm{Y} = [\mathbf{y}_1\, \mathbf{y}_2\, \cdots\, \mathbf{y}_n]^\top \qquad \mathrm{Y}_o = [\mathbf{y}_{1o}\, \mathbf{y}_{2o}\, \cdots\, \mathbf{y}_{no}]^\top . \tag{4.4.4}$$

Then (4.4.3) can be rewritten as

$$[\,\hat{\alpha},\,\hat{\boldsymbol{\theta}}\,] = \underset{\alpha,\boldsymbol{\theta}}{\operatorname{argmin}} \|\mathrm{Y} - \mathrm{Y}_o\|_F^2 \tag{4.4.5}$$

subject to

$$\mathrm{Y}_o\boldsymbol{\theta} - \alpha\mathbf{1}_n = \mathbf{0}_n \tag{4.4.6}$$

where $\mathbf{1}_n(\mathbf{0}_n)$ is a vector in $\mathcal{R}^n$ of all ones (zeros), and $\|\mathrm{A}\|_F$ is the Frobenius norm of the matrix A.

The parameter α is eliminated next. The data is centered by using the orthogonal projector matrix $\mathrm{G} = \mathrm{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$ which has the property $\mathrm{G}\mathbf{1}_n = \mathbf{0}_n$. It is easy to verify that

$$\widetilde{\mathrm{Y}} = \mathrm{GY} = [\widetilde{\mathbf{y}}_1\, \widetilde{\mathbf{y}}_2\, \cdots\, \widetilde{\mathbf{y}}_n]^\top \qquad \widetilde{\mathbf{y}}_i = \mathbf{y}_i - \frac{1}{n}\sum_{i=1}^{n}\mathbf{y}_i = \mathbf{y}_i - \bar{\mathbf{y}} . \tag{4.4.7}$$

The matrix $\widetilde{\mathrm{Y}}_o = \mathrm{GY}_o$ is similarly defined. The parameter estimate $\hat{\boldsymbol{\theta}}$ is then obtained from the minimization

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \|\widetilde{\mathrm{Y}} - \widetilde{\mathrm{Y}}_o\|_F^2 \tag{4.4.8}$$

subject to

$$\widetilde{\mathrm{Y}}_o \boldsymbol{\theta} = \mathbf{0}_n \ . \tag{4.4.9}$$

The constraint (4.4.9) implies that the rank of the true data matrix $\widetilde{\mathrm{Y}}_o$ is only $p-1$ and that the true $\boldsymbol{\theta}$ spans its null space. Indeed, our linear model requires that the true data points belong to a hyperplane in $\mathcal{R}^p$ which is a $(p-1)$-dimensional affine subspace. The vector $\boldsymbol{\theta}$ is the unit normal to this plane.

The available measurements, however, are located nearby the hyperplane and thus the measurement matrix $\widetilde{\mathrm{Y}}$ has full rank p. The solution of the TLS thus is the rank $p-1$ approximation of $\widetilde{\mathrm{Y}}$. This approximation is obtained from the SVD of $\widetilde{\mathrm{Y}}$ written as a dyadic sum

$$\widetilde{\mathrm{Y}} = \sum_{k=1}^{p} \widetilde{\sigma}_k \widetilde{\mathbf{u}}_k \widetilde{\mathbf{v}}_k^\top \tag{4.4.10}$$

where the singular vectors $\widetilde{\mathbf{u}}_i$, $i = 1, \ldots, n$ and $\widetilde{\mathbf{v}}_j$, $j = 1, \ldots, p$ provide orthonormal bases for the four linear subspaces associated with the matrix $\widetilde{\mathrm{Y}}$ [38, Sec.2.6.2], and $\widetilde{\sigma}_1 \geq \widetilde{\sigma}_2 \geq \cdots \geq \widetilde{\sigma}_p > 0$ are the singular values of this full rank matrix.

The optimum approximation yielding the minimum Frobenius norm for the error is the truncation of the dyadic sum (4.4.10) at $p-1$ terms [112, p.31]

$$\widehat{\widetilde{\mathrm{Y}}} = \sum_{k=1}^{p-1} \widetilde{\sigma}_k \widetilde{\mathbf{u}}_k \widetilde{\mathbf{v}}_k^\top \tag{4.4.11}$$

where the matrix $\widehat{\widetilde{\mathrm{Y}}}$ contains the centered *corrected measurements* $\widehat{\widetilde{\mathbf{y}}}$. These corrected measurements are the orthogonal projections of the available $\widetilde{\mathbf{y}}_i$ on the hyperplane characterized by the parameter estimates (Figure 4.6). The TLS estimator is also known as *orthogonal least squares*.

The rank one null space of $\widehat{\widetilde{\mathrm{Y}}}$ is spanned by $\widetilde{\mathbf{v}}_p$, the right singular vector associated with the smallest singular value $\widetilde{\sigma}_p$ of $\widetilde{\mathrm{Y}}$ [38, p.72]. Since $\widetilde{\mathbf{v}}_p$ is a unit vector

$$\hat{\boldsymbol{\theta}} = \widetilde{\mathbf{v}}_p \ . \tag{4.4.12}$$

The estimate of α is obtained by reversing the centering operation

$$\hat{\alpha} = \bar{\mathbf{y}}^\top \hat{\boldsymbol{\theta}} \ . \tag{4.4.13}$$

The parameter estimates of the linear EIV model can be also obtained in a different, though completely equivalent way. We define the carrier vector $\mathbf{x}$ by augmenting the variables with a constant

$$\mathbf{x} = [\mathbf{y}^\top \ \ -1]^\top \qquad\qquad \sigma^2 \mathrm{C} = \sigma^2 \begin{bmatrix} \mathrm{I}_p & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix} \tag{4.4.14}$$

which implies that the covariance matrix of the carriers is singular. Using the $n \times (p+1)$ matrices

$$\mathrm{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \cdots \ \mathbf{x}_n]^\top \qquad \mathrm{X}_o = [\mathbf{x}_{1o} \ \mathbf{x}_{2o} \ \cdots \ \mathbf{x}_{no}]^\top \tag{4.4.15}$$

the constraint (4.4.6) can be written as

$$\mathrm{X}_o\boldsymbol{\omega} = \mathbf{0}_n \qquad \boldsymbol{\omega} = [\boldsymbol{\theta}^\top \; \alpha]^\top \qquad \|\boldsymbol{\theta}\| = 1 \quad \alpha \geq 0 \tag{4.4.16}$$

where the subscript '1' of this parametrization in Section 4.2.6 was dropped.

Using Lagrangian multipliers it can be shown that the parameter estimate $\hat{\boldsymbol{\omega}}$ is the eigenvector of the *generalized* eigenproblem

$$\mathrm{X}^\top\mathrm{X}\boldsymbol{\omega} = \lambda\mathrm{C}\boldsymbol{\omega} \tag{4.4.17}$$

corresponding to the smallest eigenvalue λ_{min}. This eigenproblem is equivalent to the definition of the right singular values of the matrix $\widetilde{\mathrm{Y}}$ [38, Sec.8.3]. The condition $\|\hat{\boldsymbol{\theta}}\| = 1$ is then imposed on the vector $\hat{\boldsymbol{\omega}}$.

The first order approximation for the covariance of the parameter estimate is [70, Sec.5.2.2]

$$\mathrm{C}_{\hat{\boldsymbol{\omega}}} = \hat{\sigma}^2(\mathrm{X}^\top\mathrm{X} - \lambda_{min}\mathrm{C})^+ \tag{4.4.18}$$

where the pseudoinverse has to be used since the matrix has rank p following (4.4.17). The estimate of the noise standard deviation is

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^n \hat{g}(\mathbf{y}_i)^2}{n-p+1} = \frac{\lambda_{min}}{n-p+1} = \frac{\widetilde{\sigma}_p^2}{n-p+1} \tag{4.4.19}$$

where $\hat{g}(\mathbf{y}_i) = \mathbf{y}_i^\top\hat{\boldsymbol{\theta}} - \hat{\alpha}$ are the residuals. The covariances for the other parametrizations of the linear EIV model, $\boldsymbol{\omega}_2$ (4.2.53) and $\boldsymbol{\omega}_3$ (4.2.55) can be obtained through error propagation.

Note that when computing the TLS estimate with either of the two methods, special care has to be taken to execute *all* the required processing steps. The first approach starts with the data being centered, while in the second approach a generalized eigenproblem has to be solved. These steps are sometimes neglected in computer vision algorithms.

In the traditional linear regression model only the variable z is corrupted by noise (4.2.36), and the constraint is

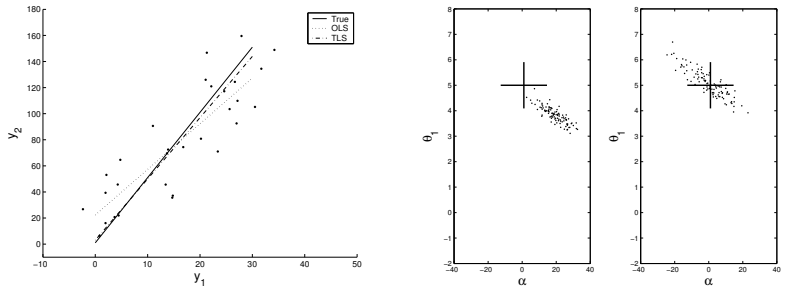
$$z_{io} = \alpha + \mathbf{x}_{io}^\top\boldsymbol{\theta} \qquad \mathbf{x}_{io} = \mathbf{x}(\mathbf{y}_{io}) \qquad i = 1, \ldots, n\,. \tag{4.4.20}$$

This model is actually valid for fewer computer vision problems (Figure 4.4) than it is used in the literature. The corresponding estimator is the well known (ordinary) *least squares* (OLS)

$$\hat{\boldsymbol{\omega}} = (\mathrm{X}_o^\top\mathrm{X}_o)^{-1}\mathrm{X}_o^\top\mathbf{z} \qquad \mathrm{C}_{\hat{\boldsymbol{\omega}}} = \hat{\sigma}^2(\mathrm{X}_o^\top\mathrm{X}_o)^{-1} \qquad \hat{\sigma}^2 = \frac{\sum_{i=1}^n z_i^2}{n-p} \tag{4.4.21}$$

where

$$\mathrm{X}_o = \begin{bmatrix} \mathbf{x}_{1o} & \mathbf{x}_{2o} & \cdots & \mathbf{x}_{no} \\ 1 & 1 & \cdots & 1 \end{bmatrix}^\top \qquad \mathbf{z} = [z_1 \; z_2 \; \cdots \; z_n]^\top\,. \tag{4.4.22}$$



(a) (b) (c)

Figure 4.16. OLS vs. TLS estimation of a linear EIV model. (a) A typical trial. (b) The scatterplot of the OLS estimates. A significant bias is present. (c) The scatterplot of the TLS estimates. The true parameter values correspond to the location marked '+'.

If the matrix X_o is poorly conditioned the pseudoinverse should be used instead of the full inverse.

In the presence of significant measurement noise, using the OLS estimator when the data obeys the full EIV model (4.4.1) results in biased estimates [112, p.232]. This is illustrated in Figure 4.16. The $n = 30$ data points are generated from the model

$$5y_{1o} - y_{2o} + 1 = 0 \qquad \delta\mathbf{y} \sim NI(\mathbf{0}, 5^2\mathrm{I}_2) \tag{4.4.23}$$

where $NI(\cdot)$ stands for independent normally distributed noise. Note that the constraint is not in the Hessian normal form but

$$\alpha + \theta_1 y_{1o} - y_{2o} = 0 \qquad \theta_1 = 5 \quad \alpha = 1 \tag{4.4.24}$$

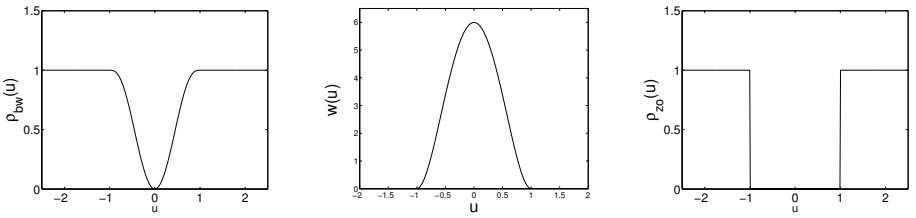
where, in order to compare the performance of the OLS and TLS estimators, the parameter θ_2 was set to -1. When the traditional regression model is associated with this data it is assumed that

$$y_{2o} \equiv z_o = \theta_1 y_{1o} + \alpha \qquad \delta z \sim NI(0, 5^2) \tag{4.4.25}$$

and the OLS estimator (4.4.21) is used to find $\hat{\theta}_1$ and $\hat{\alpha}$. The scatterplot of the result of 100 trials is shown in Figure 4.16b, and the estimates are far away from the true values.

Either TLS estimation method discussed above can be employed to find the TLS estimate. However, to eliminate the multiplicative ambiguity of the parameters the ancillary constraint $\hat{\theta}_2 = -1$ has to be used. See [112, Sec. 2.3.2]. The TLS estimates are unbiased and the scatterplot is centered on the true values (Figure 4.16c).

Throughout this section we have tacitly assumed that the data is not degenerate, i.e., the measurement matrix Y has full rank p. Both the TLS and OLS estimators can be adapted for the rank deficient case, though then the parameter estimates are no longer unique. Techniques similar to the ones described in this section yield minimum norm solutions. See [112, Chap.3] for the case of the TLS estimator.



(a) (b) (c)

Figure 4.17. Redescending M-estimators. (a) Biweight loss function. (b) The weight function for biweight. (c) Zero-one loss function.

4.4.2 M-estimators

The robust equivalent of the least squares family are the M-estimators, first proposed in 1964 by Huber as a generalization of the maximum likelihood technique in which contaminations in the data distribution are tolerated. See [67] for an introduction to M-estimators and [49] for a more in-depth discussion. We will focus only on the class of M-estimators most recommended for computer vision applications.

The robust formulation of (4.2.48) is

$$[\,\hat{\alpha},\,\hat{\boldsymbol{\theta}}\,] = \underset{\alpha,\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} \rho\left(\frac{1}{s} g(\mathbf{y}_i)\right) = \underset{\alpha,\boldsymbol{\theta}}{\operatorname{argmin}} \mathcal{J}_M \tag{4.4.26}$$

where s is a parameter which depends on σ, the (unknown) scale of the inlier noise (4.4.1). With a slight abuse of notation s will also be called scale. The loss function $\rho(u)$ satisfies the following properties: nonnegative with $\rho(0) = 0$, even symmetric $\rho(u) = \rho(-u)$, and nondecreasing with $|u|$. For $\rho(u) = u^2$ we obtain the LS objective function (4.4.3).

The different M-estimators introduced in the statistical literature differ through the distribution assumed for the data. See [5] for a discussion in the context of computer vision. However, none of these distributions will provide an accurate model in a real application. Thus, the distinctive theoretical properties of different M-estimators are less relevant in practice.

The *redescending* M-estimators are characterized by bounded loss functions

$$0 \le \rho(u) \le 1 \quad |u| \le 1 \qquad\qquad \rho(u) = 1 \quad |u| > 1\,. \tag{4.4.27}$$

As will be shown below, in a redescending M-estimator only those data points which are at distance less than s from the current fit are taken into account. This yields better outlier rejection properties than that of the M-estimators with nonredescending loss functions [69], [116].

The following class of redescending loss functions covers several important M-estimators

$$\rho(u) = \begin{cases} 1 - (1-u^2)^d & |u| \le 1 \\ 1 & |u| > 1 \end{cases} \tag{4.4.28}$$

where $d = 1, 2, 3$. The loss functions have continuous derivatives up the $(d-1)$-th order, and a unique minimum in $\rho(0) = 0$.

Tukey's *biweight* function $\rho_{bw}(u)$ (Figure 4.17a) is obtained for $d = 3$ [67, p.295]. This loss function is widely used in the statistical literature and was known at least a century before robust estimation [40, p.151]. See also [42, vol.I, p.323]. The loss function obtained for $d = 2$ will be denoted $\rho_e(u)$. The case $d = 1$ yields the *skipped mean* loss function, a name borrowed from robust location estimators [90, p.181]

$$\rho_{sm}(u) = \begin{cases} u^2 & |u| \leq 1 \\ 1 & |u| > 1 \end{cases} \tag{4.4.29}$$

which has discontinuous first derivative. It is often used in vision applications, e.g., [109].

In the objective function of any M-estimator the geometric distances (4.2.46) are normalized by the scale s. Since $\rho(u)$ is an even function we do not need to use absolute values in (4.4.26). In redescending M-estimators the scale acts as a hard rejection threshold, and thus its value is of paramount importance. For the moment we will assume that a satisfactory value is already available for s, but will return to this topic in Section 4.4.3.

The M-estimator equivalent to the total least squares is obtained following either TLS method discussed in Section 4.4.1. For example, it can be shown that instead of (4.4.17), the M-estimate of $\boldsymbol{\omega}$ (4.4.16) is the eigenvector corresponding to the the smallest eigenvalue of the generalized eigenproblem

$$\mathrm{X}^\top \mathrm{W} \mathrm{X} \boldsymbol{\omega} = \lambda \mathrm{C} \boldsymbol{\omega} \tag{4.4.30}$$

where $\mathrm{W} \in \mathcal{R}^{n \times n}$ is the diagonal matrix of the nonnegative weights

$$w_i = w(u_i) = \frac{1}{u_i} \frac{d\rho(u_i)}{du} \geq 0 \qquad u_i = \frac{\hat{g}(\mathbf{y}_i)}{s} \quad i = 1, \ldots, n \; . \tag{4.4.31}$$

Thus, in redescending M-estimators $w(u) = 0$ for $|u| > 1$, i.e., the data points whose residual $\hat{g}(\mathbf{y}_i) = \mathbf{y}_i^\top \hat{\boldsymbol{\theta}} - \hat{\alpha}$ relative to the current fit is larger than the scale threshold s are discarded from the computations. The weights $w_{bw}(u) = 6(1-u^2)^2$ derived from the biweight loss function are shown in Figure 4.17b. The weigths derived from the $\rho_e(u)$ loss function are proportional to the Epanechnikov kernel (4.3.17). For traditional regression instead of (4.4.21) the M-estimate is

$$\hat{\boldsymbol{\omega}} = (\mathrm{X}_o^\top \mathrm{W} \mathrm{X}_o)^{-1} \mathrm{X}_o^\top \mathrm{W} \mathbf{z} \; . \tag{4.4.32}$$

The residuals $\hat{g}(\mathbf{y}_i)$ in the weights w_i require values for the parameter estimates. Therefore, the M-estimates can be found only by an iterative procedure.

M-estimation with Iterative Weighted Least Squares

Given the scale s.

1. Obtain the initial parameter estimate $\hat{\boldsymbol{\omega}}^{[0]}$ with total least squares.

2. Compute the weights $w_i^{[l+1]}$, $l = 0, 1, \ldots$.

3. Obtain the updated parameter estimates, $\hat{\boldsymbol{\omega}}^{[l+1]}$.

4. Verify if $\|\hat{\boldsymbol{\omega}}^{[l+1]} - \hat{\boldsymbol{\omega}}^{[l]}\|$ is less than the tolerance. If yes, stop.

5. Replace $\hat{\boldsymbol{\omega}}^{[l]}$ with $\hat{\boldsymbol{\omega}}^{[l+1]}$. Return to Step 2.

For the traditional regression the procedure is identical. See [67, p.306]. A different way of computing linear EIV regression M-estimates is described in [116].

The objective function minimized for redescending M-estimators is not convex, and therefore the convergence to a global minimum is not guaranteed. Nevertheless, in practice convergence is always achieved [67, p.307], and if the initial fit and the chosen scale value are adequate, the obtained solution is satisfactory. These two conditions are much more influential than the precise nature of the employed loss function. Note that at every iteration all the data points regarded as inliers are processed, and thus there is no need for postprocessing, as is the case with the elemental subsets based numerical optimization technique discussed in Section 4.2.7.

In the statistical literature often the scale threshold s is defined as the product between $\hat{\sigma}$ the robust estimate for the standard deviation of the inlier noise (4.4.1) and a tuning constant. The tuning constant is derived from the asymptotic properties of the simplest location estimator, the mean [67, p.296]. Therefore its value is rarely meaningful in real applications. Our definition of redescending M-estimators avoids the problem of tuning by using the inlier/outlier classification threshold as the scale parameter s.

The case $d = 0$ in (4.4.28) yields the *zero-one* loss function

$$\rho_{zo}(u) = \begin{cases} 0 & |u| \leq 1 \\ 1 & |u| > 1 \end{cases} \tag{4.4.33}$$

shown in Figure 4.17c. The zero-one loss function is also a redescending M-estimator, however, is no longer continuous and does not have a unique minimum in $u = 0$. It is only mentioned since in Section 4.4.4 will be used to link the M-estimators to other robust regression techniques such as LMedS or RANSAC. The zero-one M-estimator is not recommended in applications. The weight function (4.4.31) is nonzero only at the boundary and the corresponding M-estimator has poor local robustness properties. That is, in a critical data configuration a single data point can have a very large influence on the parameter estimates.

4.4.3 Median Absolute Deviation Scale Estimate

Access to a reliable scale parameter s is a necessary condition for the minimization procedure (4.4.26) to succeed. The scale s is a strictly monotonically increasing function of σ the standard deviation of the inlier noise. Since σ is a nuisance parameter of the model, it can be estimated together with α and $\boldsymbol{\theta}$ at every iteration of the M-estimation process [67, p.307]. An example of a vision application employing this approach is [7]. However, the strategy is less robust than providing the main estimation process with a fixed scale value [68]. In the latter case we talk about an *M-estimator with auxiliary scale* [69], [101].

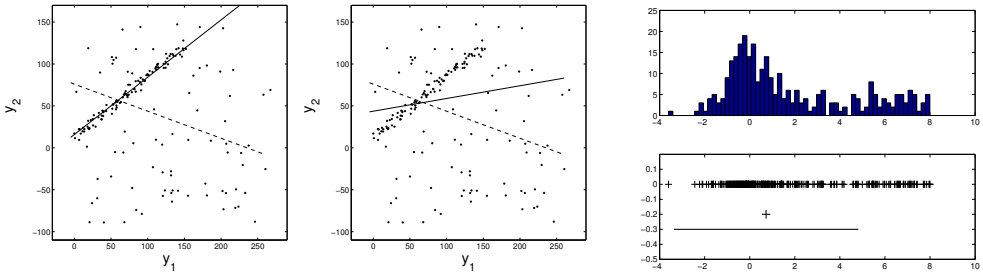


Figure 4.18. Sensitivity of the M-estimation to the $\hat{s}_{mad}$ scale value. Dashed line–initial TLS fit. Solid line–biweight M-estimate. (a) $c = 1.5$. (b) $c = 3.5$. (c) Overestimation of the scale in the presence of skewness. The median of the residuals is marked '+' under the sorted data points. The bar below corresponds to $\pm\hat{s}_{mad}$ computed with $c = 3$.

Two different approaches can be used to obtain the scale prior to the parameter estimation. It can be either *arbitrarily* set by the user, or it can be derived from the data in a pilot estimation procedure. The first approach is widely used in the robust regression techniques developed within the vision community, such as RANSAC or Hough transform. The reason is that it allows an easy way to tune a method to the available data. The second approach is often adopted in the statistical literature for M-estimators and is implicitly employed in the LMedS estimator.

The most frequently used off-line scale estimator is the *median absolute deviation* (MAD), which is based on the residuals $\hat{g}(\mathbf{y}_i)$ relative to an initial (nonrobust TLS) fit

$$\hat{s}_{mad} = c \underset{i}{\text{med}} \, |\hat{g}(\mathbf{y}_i) - \underset{j}{\text{med}} \, \hat{g}(\mathbf{y}_j)| \tag{4.4.34}$$

where c is a constant to be set by the user. The MAD scale estimate measures the spread of the residuals around their median.

In the statistical literature the constant in (4.4.34) is often taken as $c = 1.4826$. However, this value is used to obtain a consistent estimate for σ when *all the residuals* obey a normal distribution [67, p.302]. In computer vision applications where often the percentage of outliers is high, the conditions is strongly violated. In the redescending M-estimators the role of the scale parameter s is to define the inlier/outlier classification threshold. The order of magnitude of the scale can be established by computing the MAD expression, and the rejection threshold is then set as a multiple of this value. There is no need for assumptions about the residual distribution. In [106] the standard deviation of the inlier noise $\hat{\sigma}$ was computed as 1.4826 times a robust scale estimate similar to MAD, the minimum of the LMedS optimization criterion (4.4.36). The rejection threshold was set at $1.96\hat{\sigma}$ by assuming normally distributed residuals. The result is actually three times the computed MAD value, and could be obtained by setting $c = 3$ without any assumption about the distribution of the residuals.

The example in Figures 4.18a and 4.18b illustrates not only the importance of the scale

value for M-estimation but also the danger of being locked into the nature of the residuals. The data contains 100 inliers and 75 outliers, and as expected the initial TLS fit is completely wrong. When the scale parameter is set small by choosing for $\hat{s}_{mad}$ the constant $c = 1.5$, at convergence the final M-estimate is satisfactory (Figure 4.18a). When the scale $\hat{s}_{mad}$ is larger, $c = 3.5$, the optimization process converges to a local minimum of the objective function. This minimum does not correspond to a robust fit (Figure 4.18b). Note that $c = 3.5$ is about the value of the constant which would have been used under the assumption of normally distributed inlier noise.

The location estimator employed for centering the residuals in (4.4.34) is the median, while the MAD estimate is computed with the second, outer median. However, the median is a reliable estimator only when the distribution underlying the data is unimodal and symmetric [49, p.29]. It is easy to see that for a heavily skewed distribution, i.e., with a long tail on one side, the median will be biased toward the tail. For such distributions the MAD estimator severely overestimates the scale since the 50th percentile of the centered residuals is now shifted toward the boundary of the inlier distribution. The tail is most often due to outliers, and the amount of overestimation increases with both the decrease of the inlier/outlier ratio and the lengthening of the tail. In the example in Figure 4.18c the inliers (at the left) were obtained from a standard normal distribution. The median is 0.73 instead of zero. The scale computed with $c = 3$ is $\hat{s}_{mad} = 4.08$ which in much larger than 2.5, a reasonable value for the spread of the inliers. Again, c should be chosen smaller.

Scale estimators which avoid centering the data were proposed in the statistical literature [89], but they are computationally intensive and their advantage for vision applications is not immediate. We must conclude that the MAD scale estimate has to be used with care in robust algorithms dealing with real data. Whenever available, independent information provided by the problem at hand should be exploited to validate the obtained scale. The influence of the scale parameter s on the performance of M-estimators can be entirely avoided by a different approach toward this family of robust estimators. This will be discussed in Section 4.4.5.

4.4.4 LMedS, RANSAC and Hough Transform

The origin of these three robust techniques was described in Section 4.1. Now will show that they all can be expressed as M-estimators with auxiliary scale.

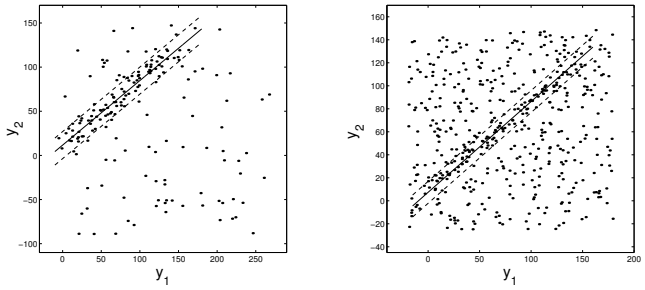
The *least median of squares* (LMedS) estimator is a least k-th order statistics estimator (4.2.22) for $k = n/2$, and is the main topic of the book [90]. The LMedS estimates are obtained from

$$[\,\hat{\alpha},\,\hat{\boldsymbol{\theta}}\,] = \underset{\alpha,\boldsymbol{\theta}}{\operatorname{argmin}} \underset{i}{\operatorname{med}}\, g(\mathbf{y}_i)^2 \tag{4.4.35}$$

but in practice we can use

$$[\,\hat{\alpha},\,\hat{\boldsymbol{\theta}}\,] = \underset{\alpha,\boldsymbol{\theta}}{\operatorname{argmin}} \underset{i}{\operatorname{med}}\, |g(\mathbf{y}_i)| \,. \tag{4.4.36}$$

The difference between the two definitions is largely theoretical, and becomes relevant only when the number of data points n is small and even, while the median is computed as the



(a) (b)

Figure 4.19. The difference between LMedS and RANSAC. (a) LMeds: finds the location of the narrowest band containing half the data. (b) RANSAC: finds the location of the densest band of width specified by the user.

average of the two central values [90, p.126]. Once the median is defined as the $[n/2]$-th order statistics, the two definitions always yield the same solution. By minimizing the median of the residuals, the LMedS estimator finds in the space of the data the narrowest cylinder which contains at least half the points (Figure 4.19a). The minimization is performed with the elemental subsets based search technique discussed in Section 4.2.7.

The scale parameter s does not appear explicitly in the above definition of the LMedS estimator. Instead of setting an upper bound on the value of the scale, the inlier/outlier threshold of the redescending M-estimator, in the LMedS a lower bound on the percentage of inliers (fifty percent) is imposed. This eliminates the need for the user to guess the amount of measurement noise, and as long as the inliers are in absolute majority, a somewhat better robust behavior is obtained. For example, the LMedS estimator will successfully process the data in Figure 4.18a.

The relation between the scale parameter and the bound on the percentage of inliers is revealed if the equivalent condition of half the data points being outside of the cylinder, is written as

$$\frac{1}{n}\sum_{i=1}^{n}\rho_{zo}\left(\frac{1}{s}g(\mathbf{y}_i)\right) = \frac{1}{2} \tag{4.4.37}$$

where ρ_{zo} is the zero-one loss function (4.4.33), and the scale parameter is now regarded as a function of the residuals $s\left[g(\mathbf{y}_1),\ldots,g(\mathbf{y}_n)\right]$. By defining $s = \underset{i}{\text{med}}\,|g(\mathbf{y}_i)|$ the LMedS estimator becomes

$$[\,\hat{\alpha},\,\hat{\boldsymbol{\theta}}\,] = \underset{\alpha,\boldsymbol{\theta}}{\text{argmin}}\; s\left[g(\mathbf{y}_1),\ldots,g(\mathbf{y}_n)\right] \qquad \text{subject to (4.4.37).} \tag{4.4.38}$$

The new definition of LMedS is a particular case of the *S-estimators*, which while popular in statistics, are not widely know in the vision community. For an introduction to S-estimators see [90, pp.135–143], and for a more detailed treatment in the context of EIV

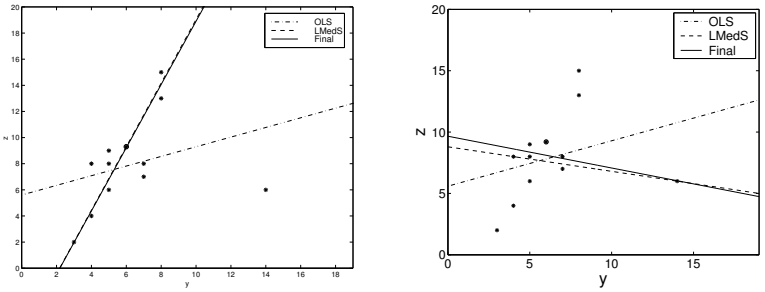


Figure 4.20. The poor local robustness of the LMedS estimator. The difference between the data sets in (a) and (b) is that the point $(6,\ 9.3)$ was moved to $(6,\ 9.2)$.

models [116]. Let $\hat{s}$ be the minimum of s in (4.4.38). Then, it can be shown that

$$[\,\hat{\alpha},\,\hat{\boldsymbol{\theta}}\,] = \underset{\alpha,\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} \rho_{zo}\left(\frac{1}{\hat{s}}g(\mathbf{y}_i)\right) \tag{4.4.39}$$

and thus the S-estimators are in fact M-estimators with auxiliary scale.

The value of $\hat{s}$ can also be used as a scale estimator for the noise corrupting the inliers. All the observations made in Section 4.4.3 remain valid. For example, when the inliers are no longer the absolute majority in the data the LMedS fit is incorrect, and the residuals used to compute $\hat{s}$ are not reliable.

The *Random Sample Consensus* (RANSAC) estimator predates the LMedS [26]. Since the same elemental subsets based procedure is used to optimize their objective function, sometimes the two techniques were mistakenly considered as being very similar, e.g., [75]. However, their similarity should be judged examining the objective functions and not the way the optimization is implemented. In LMedS the scale is computed from a condition set on the percentage of inliers (4.4.38). In RANSAC the following minimization problem is solved

$$[\,\hat{\alpha},\,\hat{\boldsymbol{\theta}}\,] = \underset{\alpha,\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} \rho_{zo}\left(\frac{1}{\hat{s}}g(\mathbf{y}_i)\right) \qquad \text{given } \hat{s} \tag{4.4.40}$$

that is, the scale is *provided* by the user. This is a critical difference. Note that (4.4.40) is the same as (4.2.30). Since it is relative easy to tune RANSAC to the data, it can also handle situations in which LMedS would already fail due to the large percentage of outliers (Figure 4.19b). Today RANSAC replaced LMedS in most vision applications, e.g., [65], [84], [108].

The use of the zero-one loss function in both LMedS and RANSAC yields very poor local robustness properties, as it is illustrated in Figure 4.20, an example inspired by [2]. The $n = 12$ data points appear to be a simple case of robust linear regression for which the traditional regression model (4.2.37) was used. The single outlier on the right corrupts the least squares (OLS) estimator. The LMedS estimator, however, succeeds to recover

the correct fit (Figure 4.20a), and the ordinary least squares postprocessing of the points declared inliers (Final), does not yield any further change. The data in Figure 4.20b seems to be the same but now the LMedS, and therefore the postprocessing, completely failed. Actually the difference between the two data sets is that the point $(6,\ 9.3)$ was moved to $(6,\ 9.2)$.

The configuration of this data, however, is a critical one. The six points in the center can be grouped either with the points which appear to be also inliers (Figure 4.20a), or with the single outlier on the right (Figure 4.20b). In either case the grouping yields an absolute majority of points which is preferred by LMedS. There is a hidden bimodality in the data, and as a consequence a delicate equilibrium exist between the correct and the incorrect fit.

In this example the LMedS minimization (4.4.36) seeks the narrowest band containing at least six data points. The width of the band is measured along the z axis, and its boundary is always defined by two of the data points [90, p.126]. This is equivalent to using the zero-one loss function in the optimization criterion (4.4.39). A small shift of one of the points thus can change to which fit does the value of the minimum in (4.4.36) correspond to. The instability of the LMedS is discussed in a practical setting in [45], while more theoretical issues are addressed in [23]. A similar behavior is also present in RANSAC due to (4.4.40).

For both LMedS and RANSAC several variants were introduced in which the zero-one loss function is replaced by a smooth function. Since then more point have nonzero weights in the optimization, the local robustness properties of the estimators improve. The *least trimmed squares* (LTS) estimator [90, p.132]

$$[\,\hat{\alpha},\ \hat{\boldsymbol{\theta}}\,] = \underset{\alpha,\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^{k} g(\mathbf{y})^2_{i:n} \tag{4.4.41}$$

minimizes the sum of squares of the k smallest residuals, where k has to be provided by the user. Similar to LMedS, the absolute values of the residuals can also be used.

In the first smooth variant of RANSAC the zero-one loss function was replaced with the skipped mean (4.4.29), and was called MSAC [109]. Recently the same loss function was used in a maximum a posteriori formulation of RANSAC, the MAPSAC estimator [105]. A maximum likelihood motivated variant, the MLESAC [107], uses a Gaussian kernel for the inliers. Guided sampling is incorporated into the IMPSAC version of RANSAC [105]. In every variant of RANSAC the user has to provide a reasonably accurate scale value for a satisfactory performance.

The use of zero-one loss function is not the only (or main) cause of the failure of LMedS (or RANSAC). In Section 4.4.7 we show that there is a more general problem in applying robust regression methods to multistructured data.

The only robust method designed to handle multistructured data is the *Hough transform*. The idea of Hough transform is to replace the regression problems in the input domain with location problems in the space of the parameters. Then, each significant mode in the parameter space corresponds to an instance of the model in the input space. There is a huge literature dedicated to every conceivable aspect of this technique. The survey papers [50], [62], [82] contain hundreds of references.

Since we are focusing here on the connection between the redescending M-estimators

and the Hough transform, only the *randomized* Hough transform (RHT) will be considered [56]. Their equivalence is the most straightforward, but the same equivalence also exists for all the other variants of the Hough transform as well. The feature space in RHT is built with elemental subsets, and thus we have a mapping from p data points to a point in the parameter space.

Traditionally the parameter space is quantized into bins, i.e., it is an accumulator. The bins containing the largest number of votes yield the parameters of the significant structures in the input domain. This can be described formally as

$$[\,\hat{\alpha},\,\hat{\boldsymbol{\beta}}\,]_k = \underset{[\alpha,\boldsymbol{\beta}]}{\operatorname{argmax}}_k \frac{1}{n}\sum_{i=1}^{n} \kappa_{zo}\left(s_\alpha, s_{\beta_1}, \ldots, s_{\beta_{p-1}}; g(\mathbf{y}_i)\right) \tag{4.4.42}$$

where $\kappa_{zo}(u) = 1 - \rho_{zo}(u)$ and $s_\alpha, s_{\beta_1}, \ldots, s_{\beta_{p-1}}$ define the size (scale) of a bin along each parameter coordinate. The index k stands for the different local maxima. Note that the parametrization uses the polar angles as discussed in Section 4.2.6.

The definition (4.4.42) is that of a redescending M-estimator with auxiliary scale, where the criterion is a maximization instead of a minimization. The accuracy of the scale parameters is a necessary condition for a satisfactory performance, an issue widely discussed in the Hough transform literature. The advantage of distributing the votes around adjacent bins was recognized early [102]. Later the equivalence with M-estimators was also identified, and the zero-one loss function is often replaced with a continuous function [61], [60], [80].

In this section we have shown that all the robust techniques popular in computer vision can be reformulated as M-estimators. In Section 4.4.3 we have emphasized that the scale has a crucial influence on the performance of M-estimators. In the next section we remove this dependence by approaching the M-estimators in a different way.

4.4.5 The pbM-estimator

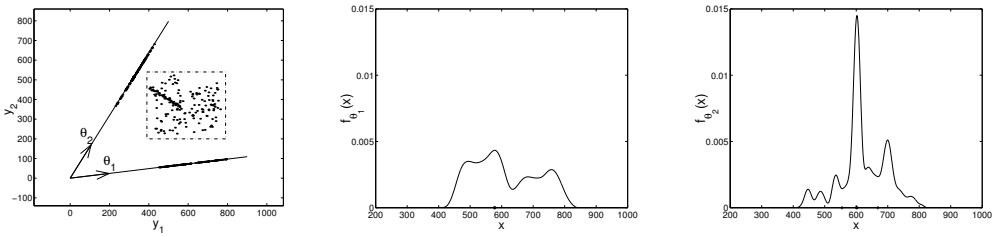
The minimization criterion (4.4.26) of the M-estimators is rewritten as

$$[\,\hat{\alpha},\,\hat{\boldsymbol{\theta}}\,] = \underset{\alpha,\boldsymbol{\theta}}{\operatorname{argmax}} \frac{1}{n}\sum_{i=1}^{n} \kappa\left(\frac{\mathbf{y}_i^\top\boldsymbol{\theta} - \alpha}{s}\right) \qquad \kappa(u) = c_\rho[1-\rho(u)] \tag{4.4.43}$$

where $\kappa(u)$ is called the *M-kernel function*. Note that for a redescending M-estimator $\kappa(u) = 0$ for $|u| > 1$ (4.4.27). The positive normalization constant c_ρ assures that $\kappa(u)$ is a proper kernel (4.3.6).

Consider the unit vector $\boldsymbol{\theta}$ defining a line through the origin in $\mathcal{R}^p$. The projections of the n data points $\mathbf{y}_i$ on this line have the one-dimensional (intrinsic) coordinates $x_i = \mathbf{y}_i^\top\boldsymbol{\theta}$. Following (4.3.5) the density of the set of points x_i, $i = 1, \ldots, n$, estimated with the kernel $K(u)$ and the bandwidth $\hat{h}_{\boldsymbol{\theta}}$ is

$$\hat{f}_{\boldsymbol{\theta}}(x) = \frac{1}{n\hat{h}_{\boldsymbol{\theta}}}\sum_{i=1}^{n} K\left(\frac{\mathbf{y}_i^\top\boldsymbol{\theta} - x}{\hat{h}_{\boldsymbol{\theta}}}\right) . \tag{4.4.44}$$



(a) (b) (c)

Figure 4.21. M-estimation through projection pursuit. When the data in the rectangle is projected orthogonally on different directions (a), the mode of the estimated density is smaller for an arbitrary direction (b), than for the direction of the normal to the linear structure (c).

Comparing (4.4.43) and (4.4.44) we can observe that if $\kappa(u)$ is taken as the kernel function, and $\hat{h}_{\boldsymbol{\theta}}$ is substituted for the scale s, the M-estimation criterion becomes

$$\hat{\boldsymbol{\theta}} \;=\; \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left[\hat{h}_{\boldsymbol{\theta}} \max_{x} \hat{f}_{\boldsymbol{\theta}}(x) \right] \tag{4.4.45}$$

$$\hat{\alpha} \;=\; \underset{x}{\operatorname{argmax}}\, \hat{f}_{\hat{\boldsymbol{\theta}}}(x) \; . \tag{4.4.46}$$

Given the M-kernel $\kappa(u)$, the bandwidth parameter $\hat{h}_{\boldsymbol{\theta}}$ can be estimated from the data according to (4.3.14). Since, as will be shown below, the value of the bandwidth has a weak influence on the the result of the M-estimation, for the entire family of redescending loss functions (4.4.28) we can use

$$\hat{h}_{\boldsymbol{\theta}} = n^{-1/5} \; \underset{i}{\operatorname{med}} \left| \, \mathbf{y}_i^{\top} \boldsymbol{\theta} - \underset{j}{\operatorname{med}} \, \mathbf{y}_j^{\top} \boldsymbol{\theta} \, \right| . \tag{4.4.47}$$

The MAD estimator is employed in (4.4.47) but its limitations (Section 4.4.3) are of less concern in this context. Also, it is easy to recognize when the data is not corrupted since the MAD expression becomes too small. In this case, instead of the density estimation most often a simple search over the projected points suffices.

The geometric interpretation of the new definition of M-estimators is similar to that of the LMedS and RANSAC techniques shown in Figure 4.19. The closer is the projection direction to the normal of the linear structure, the tighter are grouped the projected inliers together which increases the mode of the estimated density (Figure 4.21). Again a cylinder having the highest density in the data has to be located. The new approach is called *projection based* M-estimator, or pbM-estimator.

The relations (4.4.45) and (4.4.46) are the projection pursuit definition of an M-estimator. Projection pursuit was proposed by Friedman and Tukey in 1974 [30] to solve data analysis problems by seeking "interesting" low-dimensional projections of the multidimensional data. The informative value of a projection is measured with a *projection index*, such as the quantity inside the brackets in (4.4.45). The papers [48] [54] survey all the related topics.

It should be emphasized that in the projection pursuit literature the name projection pursuit regression refers to a technique different from ours. There, a nonlinear additive model is estimated by adding a new term to the model after each iteration, e.g., [44, Sec.11.2].

When in the statistical literature a linear regression problem is solved through projection pursuit, either nonrobustly [20], or robustly [90, p.143], the projection index is a scale estimate. Similar to the S-estimators the solution is obtained by minimizing the scale, now over the projection directions. The robust scale estimates, like the MAD (4.4.34) or the median of the absolute value of the residuals (4.4.38), however, have severe deficiencies for skewed distributions, as was discussed in Section 4.4.3. Thus, their use as projection index will not guarantee a better performance than that of the original implementation of the regression technique.

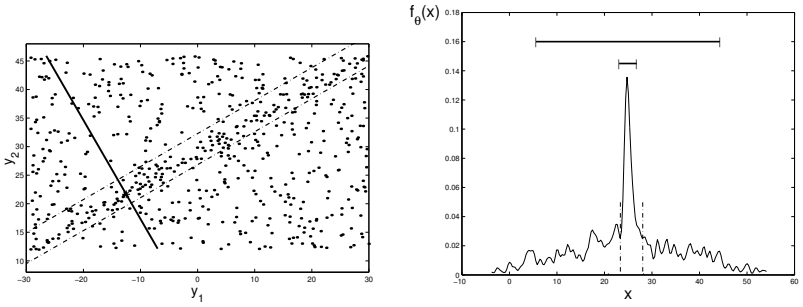
Projections were employed before in computer vision. In [81] a highly accurate implementation of the Hough transform was achieved by using local projections of the pixels onto a set of directions. Straight edges in the image were then found by detecting the maxima in the numerically differentiated projections. The L_2E estimator, proposed recently in the statistical literature [92], solves a minimization problem similar to the kernel density estimate formulation of M-estimators, however, the focus is on the parametric model of the inlier residual distribution.

The critical parameter of the redescending M-estimators is the scale s, the inlier/outlier selection threshold. The novelty of the pbM-estimator is the way the scale parameter is manipulated. The pbM-estimator avoids the need of M-estimators for an accurate scale prior to estimation by using the bandwidth $\hat{h}_{\boldsymbol{\theta}}$ as scale during the search for the optimal projection direction. The bandwidth being an approximation of the AMISE optimal solution (4.3.13) tries to preserve the sensitivity of the density estimation process as the number of data points n becomes large. This is the reason for the $n^{-1/5}$ factor in (4.4.47). Since $\hat{h}_{\boldsymbol{\theta}}$ is the outlier rejection threshold at this stage, a too small value increases the probability of assigning incorrectly the optimal projection direction to a local alignment of points. Thus, it is recommended that once n becomes large, say $n > 10^3$, the computed bandwith value is slightly increased by a factor which is monotonic in n.

After the optimal projection direction $\hat{\boldsymbol{\theta}}$ was found, the actual inlier/outlier dichotomy of the data is defined by analyzing the shape of the density around the mode. The nearest local minima on the left and on the right correspond in $\mathcal{R}^p$, the space of the data, to the transition between the inliers belonging to the sought structure (which has a higher density) and the background clutter of the outliers (which has a lower density). The locations of the minima define the values $\alpha_1 < \alpha_2$. Together with $\hat{\boldsymbol{\theta}}$ they yield the two hyperplanes in $\mathcal{R}^p$ separating the inliers from the outliers. Note that the equivalent scale of the M-estimator is $s = \alpha_2 - \alpha_1$, and that the minima may not be symmetrically located relative to the mode.

The 2D data in the example in Figure 4.22a contains 100 inliers and 500 outliers. The density of the points projected on the direction of the true normal (Figure 4.22b) has a sharp mode. Since the pbM-estimator deals only with one-dimensional densities, there is no need to use the mean shift procedure (Section 4.3.3) to find the modes, and a simple heuristic suffices to define the local minima if they are not obvious.

The advantage of the pbM-estimator arises from using a more adequate scale in the optimization. In our example, the $\hat{s}_{mad}$ scale estimate based on the TLS initial fit (to the



(a) (b)

Figure 4.22. Determining the inlier/outlier dichotomy through the density of the projected data. (a) 2D data. Solid line: optimal projection direction. Dashed lines: boundaries of the detected inlier region. (b) The kernel density estimate of the projected points. Vertical dashed lines: the left and right local minima. The bar at the top is the scale $\pm\hat{s}_{mad}$ computed with $c = 3$. The bar below is $\pm\hat{h}_{\hat{\boldsymbol{\theta}}}$, the size of the kernel support. Both are centered on the mode.

whole data) and computed with $c = 3$, is about ten times larger than $\hat{h}_{\hat{\boldsymbol{\theta}}}$, the bandwith computed for the optimal projection direction (Figure 4.22b). When a redescending M-estimator uses $\hat{s}_{mad}$, the optimization of the objective function is based on a too large band, which almost certainly leads to a nonrobust behavior.

Sometimes the detection of the minima can be fragile. See the right minimum in Figure 4.22b. A slight change in the projected location of a few data points could have changed this boundary to the next, much more significant local minimum. However, this sensitivity is tolerated by pbM-estimator. First, by the nature of the projection pursuit many different projections are investigated and thus it is probable that at least one satisfactory band is found. Second, from any reasonable inlier/outlier dichotomy of the data postprocessing of the points declared inliers (the region bounded by the two hyperplanes in $\mathcal{R}^p$) can recover the correct estimates. Since the *true* inliers are with high probability the absolute majority among the points *declared* inliers, the robust LTS estimator (4.4.41) can now be used.

The significant improvement in outlier tolerance of the pbM-estimator was obtained at the price of replacing the iterative weighted least squares algorithm of the traditional M-estimation with a search in $\mathcal{R}^p$ for the optimal projection direction $\hat{\boldsymbol{\theta}}$. This search can be efficiently implemented using the simplex based technique discussed in Section 4.2.7.

A randomly selected p-tuple of points (an elemental subset) defines the projection direction $\boldsymbol{\theta}$, from which the corresponding polar angles $\boldsymbol{\beta}$ are computed (4.2.52). The vector $\boldsymbol{\beta}$ is the first vertex of the initial simplex in $\mathcal{R}^{p-1}$. The remaining $p-1$ vertices are then defined as

$$\boldsymbol{\beta}_k = \boldsymbol{\beta} + \mathbf{e}_k * \boldsymbol{\gamma} \qquad k = 1, \ldots, (p-1) \tag{4.4.48}$$

where $\mathbf{e}_k \in R^{p-1}$ is a vector of 0-s except a 1 in the k-th element, and $\boldsymbol{\gamma}$ is a small angle. While the value of $\boldsymbol{\gamma}$ can depend on the dimension of the space, using a constant value such

as $\gamma = \pi/12$, seems to suffice in practice. Because $\boldsymbol{\theta}$ is only a projection direction, during the search the polar angles are allowed to wander outside the limits assuring a unique mapping in (4.2.52). The simplex based maximization of the projection index (4.4.45) does not have to be extremely accurate, and the number of iterations in the search should be relative small.

The projection based implementation of the M-estimators is summarized below.

The pbM-estimator

- Repeat N times:
  1. choose an elemental subset (p-tuple) by random sampling;
  2. compute the TLS estimate of $\boldsymbol{\theta}$;
  3. build the initial simplex in the space of polar angles $\boldsymbol{\beta}$;
  4. perform a simplex based direct search to find the local maximum of the projection index.
- Find the left and right local minima around the mode of the density corresponding to the largest projection index.
- Define the inlier/outlier dichotomy of the data. Postprocess the inliers to find the final estimates of α and $\boldsymbol{\theta}$.

4.4.6 Applications

The superior outlier tolerance of the pbM-estimator relative to other robust techniques is illustrated with two experiments. The percentage of inliers in the data is assumed unknown and can be significantly less than that of the outliers. Therefore the LMedS estimator cannot be applied. It is shown in [107] that MLESAC and MSAC have very similar performance and are superior to RANSAC. We have compared RANSAC and MSAC with the pbM-estimator.

In both experiments ground truth was available, and the *true* standard deviation of the inliers σ_t could be computed. The output of any robust regression is the inlier/outlier dichotomy of the data. Let the standard deviation of the points *declared* inliers measured relative to the *true fit* be $\hat{\sigma}_{in}$. The performance of the different estimators was compared through the ratio $\hat{\sigma}_{in}/\sigma_t$. For a satisfactory result this ratio should be very close to one.

The same number of computational units is used for all the techniques. A computational unit is either processing of one elemental subset (RANSAC), or one iteration in the simplex based direct search (pbM). The number of iterations in a search was restricted to 25, but often it ended earlier. Thus, the amount of computations attributed to the pbM-estimator is an upper bound.

In the *first experiment* the synthetic data contained 100 inlier points obeying an eight-dimensional linear EIV regression model (4.4.2). The measurement noise was normally distributed with covariance matrix $5^2\mathrm{I}_8$. A variable percentage of outliers was uniformly distributed within the bounding box of the region occupied in R^8 by the inliers. The number of computational units was 5000, i.e., RANSAC used 5000 elemental subsets while the

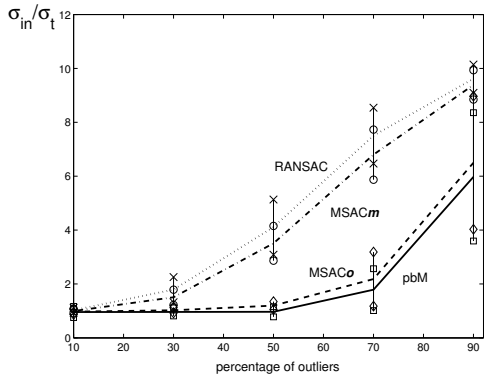


Figure 4.23. RANSAC vs. pbM-estimator. The relative standard deviation of the residuals function of the percentage of outliers. Eight dimensional synthetic data. The employed scale threshold: RANSAC – $\hat{s}_{mad}$; MSACm – $\hat{s}_{mad}$; MSACo – s_{opt}. The pbM-estimator has no tuning parameter. The vertical bars mark one standard deviation from the mean.

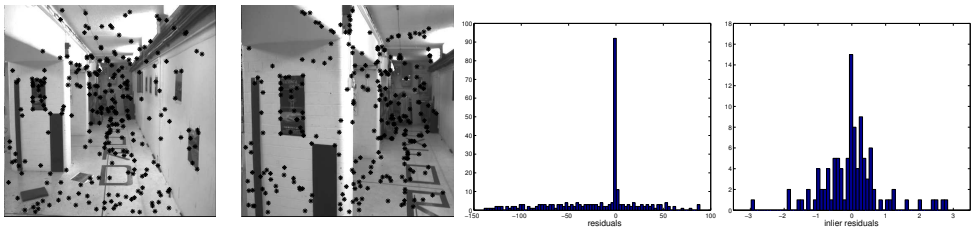
pbM-estimator initiated 200 local searches. For each experimental condition 100 trials were run. The *true sample* standard deviation of the inliers σ_t, was computed in each trial.

The scale provided to RANSAC was the $\hat{s}_{mad}$, based on the TLS fit to the data and computed with $c = 3$. The same scale was used for MSAC. However, in an optimal setting MSAC was also run with the scale $s_{opt} = 1.96\sigma_t$. Note that this information is not available in practice! The graphs in Figure 4.23 show that for any percentage of outliers the pbM-estimator performs at least as well as MSAC tuned to the optimal scale. This superior performance is obtained in a completely unsupervised fashion. The only parameters used by the pbM-estimator are the generic normalized amplitude values needed for the definition of the local minima. They do not depend on the data or on the application.

In the *second experiment,* two far apart frames from the *corridor* sequence (Figures 4.24a and 4.24b) were used to estimate the epipolar geometry from point correspondences. As was shown in Section 4.2.5 this is a nonlinear estimation problem, and therefore the role of a robust regression estimator based on the linear EIV model is restricted to selecting the correct matches. Subsequent use of a nonlinear (and nonrobust) method can recover the unbiased estimates. Several such methods are discussed in [118].

The Harris corner detector [111, Sec.4.3] was used to establish the correspondences, from which 265 point pairs were retained. The histogram of the residuals computed as orthogonal distances from the ground truth plane in 8D, is shown in Figure 4.24c. The 105 points in the central peak of the histogram were considered the inliers (Figure 4.24d). Their standard deviation was $\sigma_t = 0.88$.

The number of computational units was 15000, i.e., the pbM-estimator used 600 searches. Again, MSAC was tuned to either the optimal scale s_{opt} or to the scale derived from the MAD estimate, $\hat{s}_{mad}$. The number true inliers among the points selected by an estimator and the ratio between the standard deviation of the selected points and that of the true inlier



(a) (b) (c) (d)

Figure 4.24. Estimating the epipolar geometry for two frames of the *corridor* sequence. (a) and (b) The input images with the points used for correspondences marked. (c) Histogram of the residuals from the ground truth. (d) Histogram of the inliers.

noise are shown in the table below.

	selected points/true inliers	$\hat{\sigma}_{in}/\sigma_t$
MSAC (s_{mad})	219/105	42.32
MSAC (s_{opt})	98/87	1.69
pbM	95/88	1.36

The pbM-estimator succeeds to recover the data of interest, and behaves like an optimally tuned technique from the RANSAC family. However, in practice the tuning information is not available.

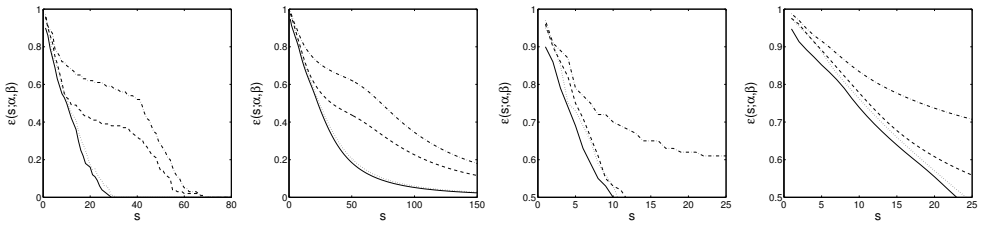
4.4.7 Structured Outliers

The problem of multistructured data is not considered in this chapter, but a discussion of robust regression cannot be complete without mentioning the issue of structured outliers. This is a particular case of multistructured data, when only two structures are present and the example shown in Figure 4.3 is a typical case. For such data, once the measurement noise becomes significant all the robust techniques, M-estimators (including the pbM-estimator), LMedS and RANSAC behave similarly to the nonrobust least squares estimator. This was first observed for the LMedS [78], and was extensively analyzed in [98]. Here we describe a more recent approach [9].

The true structures in Figure 4.3b are horizontal lines. The lower one contains 60 points and the upper one 40 points. Thus, a robust regression method should return the lower structure as inliers. The measurement noise was normally distributed with covariance $\sigma^2 \mathrm{I}_2$, $\sigma = 10$. In Section 4.4.4 it was shown that all robust techniques can be regarded as M-estimators. Therefore we consider the expression

$$\epsilon(s;\alpha,\beta) = \frac{1}{n}\sum_{i=1}^{n} \rho\left(\frac{y_1 cos\beta + y_2 sin\beta - \alpha}{s}\right) \tag{4.4.49}$$

which defines a family of curves parameterized in α and β of the line model (4.2.54) and



(a) (b) (c) (d)

Figure 4.25. Dependence of $\epsilon(s;\alpha,\beta)$ on the scale s for the data in Figure 4.3b. (a) Zero-one loss function. (b) Biweight loss function. (c) The top-left region of (a). (d) The top-left region of (b). Solid line – envelope $\epsilon_{min}(s)$. Dashed line – true parameters of the lower structure. Dotdashed line – true parameters of the upper structure. Dotted line – least squares fit parameters.

in the scale s. The envelope of this family

$$\epsilon_{min}(s) = \min_{\alpha,\beta} \epsilon(s;\alpha,\beta) \tag{4.4.50}$$

represents the value of the M-estimation minimization criterion (4.4.26) as a function of scale.

By definition, for a given value of s the curve $\epsilon(s;\alpha,\beta)$ can be only above (or touching) the envelope. The comparison of the envelope with a curve $\epsilon(s;\alpha,\beta)$ describes the relation between the employed α,β and the parameter values minimizing (4.4.49). Three sets of line parameters were investigated using the zero-one (Figure 4.25a) and the biweigth (Figure 4.25b) loss functions: the true parameters of the two structures ($\alpha = 50, 100$; $\beta = \pi/2$), and the least squares parameter estimates ($\alpha_{LS},\ \beta_{LS}$). The LS parameters yield a line similar to the one in Figure 4.3b, a nonrobust result.

Consider the case of zero-one loss function and the parameters of the lower structure (dashed line in Figure 4.25a). For this loss function $\epsilon(s;50,\pi/2)$ is the percentage of data points outside the horizontal band centered on $y_2 = 50$ and with half-width s. As expected the curve has a plateau around $\epsilon = 0.4$ corresponding to the band having one of its boundaries in the transition region between the two structures. Once the band extends into the second structure $\epsilon(s;50,\pi/2)$ further decreases. The curve, however, is not only always above the envelope, but most often also above the curve $\epsilon(s;\alpha_{LS},\beta_{LS})$. See the magnified area of small scales in Figure 4.25c.

For a given value of the scale (as in RANSAC) a fit similar to least squares will be prefered since it yields a smaller value for (4.4.49). The measurement noise being large, a band containing half the data (as in LMedS) corresponds to a scale $s > 12$, the value around which the least squares fit begins to dominate the optimization (Figure 4.25a). As a result the LMedS will always fail (Figure 4.3b). Note also the very narrow range of scale values (around $s = 10$) for which $\epsilon(s;50,\pi/2)$ is below $\epsilon(s;\alpha_{LS},\beta_{LS})$. It shows how accurately has the user to tune an estimator in the RANSAC family for a satisfactory performance.

The behavior for the biweight loss function is identical, only the curves are smoother

due to the weigthed averages (Figures 4.25b and 4.25d). When the noise corrupting the structures is small, in Figure 4.3a it is $\sigma = 2$, the envelope and the curve $\epsilon(s; 50, \pi/2)$ overlap for $s < 8$ which suffices for the LMedS criterion. See [9] for details.

We can conclude that multistructured data has to be processed first by breaking it into parts in which one structure dominates. The technique in [8] combines several of the procedures discussed in this chapter. The sampling was guided by local data density, i.e., it was assumed that the structures and the background can be roughly separated by a global threshold on nearest neighbor distances. The pbM-estimator was employed as the estimation module, and the final parameters were obtained by applying adaptive mean shift to a feature space. The technique had a Hough transform flavor, though no scale parameters were required. The density assumption, however, may fail when the structures are defined by linearizing a nonlinear problem, as it is often the case in 3D vision. Handling such multistructured data embedded in a significant background clutter, remains an open question.

4.5 Conclusion

Our goal in this chapter was to approach robust estimation from the point of view of a practitioner. We have used a common statistical framework with solid theoretical foundations to discuss the different types and classes of robust estimators. Therefore, we did not dwell on techniques which have an excellent robust behavior but are of a somewhat ad-hoc nature. These techniques, such as tensor voting [73], can provide valuable tools for solving difficult computer vision problems.

Another disregarded topic was the issue of diagnosis. Should an algorithm be able to determine its own failure, one can already talk about robust behavior. When in the late 1980's robust methods became popular in the vision community, the paper [28] was often considered as the first robust work in the vision literature. The special issue [94] and the book [6] contain representative collections of papers for the state-of-the-art today.

We have emphasized the importance of embedding into the employed model the least possible amount of assumptions necessary for the task at hand. In this way the developed algorithms are more suitable for vision applications, where the data is often more complex than in the statistical literature. However, there is a tradeoff to satisfy. As the model becomes less committed (more nonparametric), its power to extrapolate from the available data also decreases. How much is modeled rigorously and how much is purely data driven is an important decision of the designer of an algorithm. The material presented in this chapter was intended to help in taking this decision.

Acknowledgements

I must thank to several of my current and former graduate students whose work is directly or indirectly present on every page: Haifeng Chen, Dorin Comaniciu, Bogdan Georgescu, Yoram Leedan and Bogdan Matei. Long discussions with Dave Tyler from the Statistics Department, Rutgers University helped to crystallize many of the ideas described in this paper. Should they be mistaken, the blame is entirely mine. Preparation of the material was supported by the National Science Foundation under the grant IRI 99-87695.

BIBLIOGRAPHY

[1] J. Addison. Pleasures of imagination. *Spectator*, 6, No. 411, June 21, 1712.

[2] G. Antille and H. El May. The use of slices in the LMS and the method of density slices: Foundation and comparison. In Y. Dodge and J. Whittaker, editors, *Proc. 10th Symp. Computat. Statist., Neuchatel*, volume I, pages 441–445. Physica-Verlag, 1992.

[3] T. Arbel and F. P. Ferrie. On sequential accumulation of evidence. *Intl. J. of Computer Vision*, 43:205–230, 2001.

[4] P. J. Besl, J. B. Birch, and L. T. Watson. Robust window operators. In *Proceedings of the 2nd International Conference on Computer Vision*, pages 591–600, Tampa, FL, December 1988.

[5] M.J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Intl. J. of Computer Vision*, 19:57–91, 1996.

[6] K. J. Bowyer and P. J. Phillips, editors. *Empirical evaluation techniques in computer vision*. IEEE Computer Society, 1998.

[7] K. L. Boyer, M. J. Mirza, and G. Ganguly. The robust sequential estimator: A general approach and its application to surface organization in range data. *IEEE Trans. Pattern Anal. Machine Intell.*, 16:987–1001, 1994.

[8] H. Chen and P. Meer. Robust computer vision through kernel density estimation. In *Proc. European Conf. on Computer Vision,* Copenhagen, Denmark, volume I, pages 236–250, May 2002.

[9] H. Chen, P. Meer, and D. E. Tyler. Robust regression for data with multiple structures. In *2001 IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pages 1069–1075, Kauai, HI, December 2001.

[10] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Machine Intell.*, 17:790–799, 1995.

[11] E. Choi and P. Hall. Data sharpening as a prelude to density estimation. *Biometrika*, 86:941–947, 1999.

[12] W. Chojnacki, M. J. Brooks, A. van den Hengel, and D. Gawley. On the fitting of surfaces to data with covariances. *IEEE Trans. Pattern Anal. Machine Intell.*, 22:1294–1303, 2000.

[13] C.M. Christoudias, B. Georgescu, and P. Meer. Synergism in low-level vision. In *Proc. 16th International Conference on Pattern Recognition,* Quebec City, Canada, volume IV, pages 150–155, August 2002.

[14] R. T. Collins. Mean-shift blob tracking through scale space. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Madison, WI, volume II, pages 234–240, 2003.

[15] D. Comaniciu. An algorithm for data-driven bandwidth selection. *IEEE Trans. Pattern Anal. Machine Intell.*, 25:281–288, 2003.

[16] D. Comaniciu and P. Meer. Distribution free decomposition of multivariate data. *Pattern Analysis and Applications*, 2:22–30, 1999.

[17] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, 24:603–619, 2002.

[18] D. Comaniciu, V. Ramesh, and P. Meer. The variable bandwidth mean shift and data-driven scale selection. In *Proc. 8th Intl. Conf. on Computer Vision,* Vancouver, Canada, volume I, pages 438–445, July 2001.

[19] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 25:564–577, 2003.

[20] D. Donoho, I. Johnstone, P. Rousseeuw, and W. Stahel. Discussion: Projection pursuit. *Annals of Statistics*, 13:496–500, 1985.

[21] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley, second edition, 2001.

[22] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, New York, 1993.

[23] S. P. Ellis. Instability of least squares, least absolute deviation and least median of squares linear regression. *Statistical Science*, 13:337–350, 1998.

[24] O. Faugeras. *Three-Dimensional Computer Vision*. MIT Press, 1993.

[25] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *DARPA Image Understanding Workshop*, pages 71–88, University of Maryland, College Park, April 1980.

[26] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. Assoc. Comp. Mach*, 24(6):381–395, 1981.

[27] A.W. Fitzgibbon, M. Pilu, and R.B. Fisher. Direct least square fitting of ellipses. *IEEE Trans. Pattern Anal. Machine Intell.*, 21:476–480, 1999.

[28] W. Förstner. Reliability analysis of parameter estimation in linear models with applications to mensuration problems in computer vision. *Computer Vision, Graphics, and Image Processing*, 40:273–310, 1987.

[29] W. T. Freeman, E. G. Pasztor, and O. W. Carmichael. Learning in low-level vision. *Intl. J. of Computer Vision*, 40:25–47, 2000.

[30] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. Comput.*, 23:881–889, 1974.

[31] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, second edition, 1990.

[32] K. Fukunaga and L. D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Information Theory*, 21:32–40, 1975.

[33] W. Fuller. *Measurement Error Models*. Wiley, 1987.

[34] B. Georgescu and P. Meer. Balanced recovery of 3D structure and camera motion from uncalibrated image sequences. In *Proc. European Conf. on Computer Vision,* Copenhagen, Denmark, volume II, pages 294–308, 2002.

[35] B. Georgescu, I. Shimshoni, and P. Meer. Mean shift based clustering in high dimensions: A texture classification example. In *Proc. 9th Intl. Conf. on Computer Vision,* Nice, France, October 2003.

[36] E. B. Goldstein. *Sensation and Perception*. Wadsworth Publishing Co., 2nd edition, 1987.

[37] G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Number. Math.*, 14:403–420, 1970.

[38] G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins U. Press, second edition, 1989.

[39] P. Hall, T.C. Hui, and J.S. Marron. Improved variable window kernel estimates of probability densities. *Annals of Statistics*, 23:1–10, 1995.

[40] R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel. *Robust Statistics. The Approach Based on Influence Function*. Wiley, 1986.

[41] R. M. Haralick and H. Joo. 2D-3D pose estimation. In *Proceedings of the 9th International Conference on Pattern Recognition*, pages 385–391, Rome, Italy, November 1988.

[42] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*. Addison-Wesley, 1992.

[43] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[44] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[45] T. P. Hettmansperger and S. J. Sheather. A cautionary note on the method of least median of squares. *The American Statistician*, 46:79–83, 1992.

[46] P.V.C. Hough. Machine analysis of bubble chamber pictures. In *International Conference on High Energy Accelerators and Instrumentation*, Centre Européenne pour la Recherch Nucléaire (CERN), 1959.

[47] P.V.C. Hough. Method and means for recognizing complex patterns. US Patent 3,069,654, December 18, 1962.

[48] P. J. Huber. Projection pursuit (with discussion). *Annals of Statistics*, 13:435–525, 1985.

[49] P. J. Huber. *Robust Statistical Procedures*. SIAM, second edition, 1996.

[50] J. Illingworth and J. V. Kittler. A survey of the Hough transform. *Computer Vision, Graphics, and Image Processing*, 44:87–116, 1988.

[51] M. Isard and A. Blake. Condensation - Conditional density propagation for visual tracking. *Intl. J. of Computer Vision*, 29:5–28, 1998.

[52] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[53] J.M. Jolion, P. Meer, and S. Bataouche. Robust clustering with applications in computer vision. *IEEE Trans. Pattern Anal. Machine Intell.*, 13:791–802, 1991.

[54] M. C. Jones and R. Sibson. What is projection pursuit? (with discussion). *J. Royal Stat. Soc. A*, 150:1–37, 1987.

[55] B. Julesz. Early vision and focal attention. *Rev. of Modern Physics*, 63:735–772, 1991.

[56] H. Kälviäinen, P. Hirvonen, L. Xu, and E. Oja. Probabilistic and nonprobabilistic Hough transforms: Overview and comparisons. *Image and Vision Computing*, 13:239–252, 1995.

[57] K. Kanatani. Statistical bias of conic fitting and renormalization. *IEEE Trans. Pattern Anal. Machine Intell.*, 16:320–326, 1994.

[58] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier, 1996.

[59] D.Y. Kim, J.J. Kim, P. Meer, D. Mintz, and A. Rosenfeld. Robust computer vision: The least median of squares approach. In *Proceedings 1989 DARPA Image Understanding Workshop*, pages 1117–1134, Palo Alto, CA, May 1989.

[60] N. Kiryati and A. M. Bruckstein. What's in a set of points? *IEEE Trans. Pattern Anal. Machine Intell.*, 14:496–500, 1992.

[61] N. Kiryati and A. M. Bruckstein. Heteroscedastic Hough transform (HtHT): An efficient method for robust line fitting in the 'errors in the variables' problem. *Computer Vision and Image Understanding*, 78:69–83, 2000.

[62] V. F. Leavers. Survey: Which Hough transform? *Computer Vision, Graphics, and Image Processing*, 58:250–264, 1993.

[63] K.M. Lee, P. Meer, and R.H. Park. Robust adaptive segmentation of range images. *IEEE Trans. Pattern Anal. Machine Intell.*, 20:200–205, 1998.

[64] Y. Leedan and P. Meer. Heteroscedastic regression in computer vision: Problems with bilinear constraint. *Intl. J. of Computer Vision*, 37:127–150, 2000.

[65] A. Leonardis and H. Bischof. Robust recognition using eigenimages. *Computer Vision and Image Understanding*, 78:99–118, 2000.

[66] R. M. Lewis, V. Torczon, and M. W. Trosset. Direct search methods: Then and now. *J. Computational and Applied Math.*, 124:191–207, 2000.

[67] G. Li. Robust regression. In D. C. Hoaglin, F. Mosteller, and J. W. Tukey, editors, *Exploring Data Tables, Trends, and Shapes*, pages 281–343. Wiley, 1985.

[68] R. A. Maronna and V. J. Yohai. The breakdown point of simulataneous general M estimates of regression and scale. *J. of Amer. Stat. Assoc.*, 86:699–703, 1991.

[69] R. D. Martin, V. J. Yohai, and R. H. Zamar. Min-max bias robust regression. *Annals of Statistics*, 17:1608–1630, 1989.

[70] B. Matei. *Heteroscedastic Errors-in-Variables Models in Computer Vision*. PhD thesis, Department of Electrical and Computer Engineering, Rutgers University, 2001. Available at `http://www.caip.rutgers.edu/riul/research/theses.html`.

[71] B. Matei and P. Meer. Bootstrapping errors-in-variables models. In B. Triggs, A. Zisserman, and R. Szelisky, editors, *Vision Algorithms: Theory and Practice*, pages 236–252. Springer, 2000.

[72] B. Matei and P. Meer. Reduction of bias in maximum likelihood ellipse fitting. In *15th International Conference on Computer Vision and Pattern Recog.*, volume III, pages 802–806, Barcelona, Spain, September 2000.

[73] G. Medioni and P. Mordohai. The tensor voting framework. In G. Medioni and S. B. Kang, editors, *Emerging Topics in Computer Vision*. Prentice Hall, 2004.

[74] P. Meer and B. Georgescu. Edge detection with embedded confidence. *IEEE Trans. Pattern Anal. Machine Intell.*, 23:1351–1365, 2001.

[75] P. Meer, D. Mintz, D. Y. Kim, and A. Rosenfeld. Robust regression methods in computer vision: A review. *Intl. J. of Computer Vision*, 6:59–70, 1991.

[76] J. M. Mendel. *Lessons in Estimation Theory for Signal Processing, Communications, and Control*. Prentice Hall, 1995.

[77] J. V. Miller and C. V. Stewart. MUSE: Robust surface fitting using unbiased scale estimates. In *CVPR96*, pages 300–306, June 1996.

[78] D. Mintz, P. Meer, and A. Rosenfeld. Consensus by decomposition: A paradigm for fast high breakdown point robust estimation. In *Proceedings 1991 DARPA Image Understanding Workshop*, pages 345–362, La Jolla, CA, January 1992.

[79] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[80] P. L. Palmer, J. Kittler, and M. Petrou. An optimizing line finder using a Hough transform algorithm. *Computer Vision and Image Understanding*, 67:1–23, 1997.

[81] D. Petkovic, W. Niblack, and M. Flickner. Projection-based high accuracy measurement of straight line edges. *Machine Vision and Appl.*, 1:183–199, 1988.

[82] P. D. Picton. Hough transform references. *Internat. J. of Patt. Rec and Artific. Intell.*, 1:413–425, 1987.

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

[84] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *6th International Conference on Computer Vision*, pages 754–760, Bombay, India, January 1998.

[85] Z. Pylyshyn. Is vision continuous with cognition? The case for cognitive impenetrability of visual perception. *Behavioral and Brain Sciences*, 22:341–423, 1999. (with comments).

[86] S.J. Raudys and A.K. Jain. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Trans. Pattern Anal. Machine Intell.*, 13:252–264, 1991.

[87] P. J. Rousseeuw. Least median of squares regression. *J. of Amer. Stat. Assoc.*, 79:871–880, 1984.

[88] P. J. Rousseeuw. Unconventional features of positive-breakdown estimators. *Statistics & Prob. Letters*, 19:417–431, 1994.

[89] P. J. Rousseeuw and C. Croux. Alternatives to the median absolute deviation. *J. of Amer. Stat. Assoc.*, 88:1273–1283, 1993.

[90] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. Wiley, 1987.

[91] D. Ruppert and D. G. Simpson. Comment on "Unmasking Multivariate Outliers and Leverage Points", by P. J. Rousseeuw and B. C. van Zomeren. *J. of Amer. Stat. Assoc.*, 85:644–646, 1990.

[92] D. W. Scott. Parametric statistical modeling by minimum integrated square error. *Technometrics*, 43:247–285, 2001.

[93] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, 1986.

[94] Special Issue. Performance evaluation. *Machine Vision and Appl.*, 9(5/6), 1997.

[95] Special Issue. Robust statistical techniques in image understanding. *Computer Vision and Image Understanding*, 78, April 2000.

[96] G. Speyer and M. Werman. Parameter estimates for a pencil of lines: Bounds and estimators. In *Proc. European Conf. on Computer Vision,* Copenhagen, Denmark, volume I, pages 432–446, 2002.

[97] L. Stark and K.W. Bowyer. Achieving generalized object recognition through reasoning about association of function to structure. *IEEE Trans. Pattern Anal. Machine Intell.*, 13:1097–1104, 1991.

[98] C. V. Stewart. Bias in robust estimation caused by discontinuities and multiple structures. *IEEE Trans. Pattern Anal. Machine Intell.*, 19:818–833, 1997.

[99] C. V. Stewart. Robust parameter estimation in computer vision. *SIAM Reviews*, 41:513–537, 1999.

[100] C.V. Stewart. Minpran: A new robust estimator for computer vision. *IEEE Trans. Pattern Anal. Machine Intell.*, 17:925–938, 1995.

[101] K. S. Tatsuoka and D. E. Tyler. On the uniqueness of S and constrained M-functionals under non-elliptical distributions. *Annals of Statistics*, 28:1219–1243, 2000.

[102] P. R. Thrift and S. M. Dunn. Approximating point-set images by line segments using a variation of the Hough transform. *Computer Vision, Graphics, and Image Processing*, 21:383–394, 1983.

[103] A. Tirumalai and B. G. Schunk. Robust surface approximation using least median of squares. Technical Report CSE-TR-13-89, Artificial Intelligence Laboratory, 1988. University of Michigan, Ann Arbor.

[104] B. Tordoff and D.W. Murray. Guided sampling and consensus for motion estimation. In *7th European Conference on Computer Vision*, volume I, pages 82–96, Copenhagen, Denmark, May 2002.

[105] P. H. S. Torr and C. Davidson. IMPSAC: Synthesis of importance sampling and random sample consensus. *IEEE Trans. Pattern Anal. Machine Intell.*, 25:354–364, 2003.

[106] P. H. S. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *Intl. J. of Computer Vision*, 24:271–300, 1997.

[107] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156, 2000.

[108] P. H. S. Torr, A. Zisserman, and S. J. Maybank. Robust detection of degenerate configurations while estimating the fundamental matrix. *Computer Vision and Image Understanding*, 71:312–333, 1998.

[109] P.H.S. Torr and A. Zisserman. Robust computation and parametrization of multiple view relations. In *6th International Conference on Computer Vision*, pages 727–732, Bombay, India, January 1998.

[110] A. Treisman. Features and objects in visual processing. *Scientific American*, 254:114–125, 1986.

[111] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.

[112] S. Van Huffel and J. Vandewalle. *The Total Least Squares Problem. Computational Aspects and Analysis*. Society for Industrial and Applied Mathematics, 1991.

[113] M. P. Wand and M.C. Jones. *Kernel Smoothing*. Chapman & Hall, 1995.

[114] I. Weiss. Line fitting in a noisy image. *IEEE Trans. Pattern Anal. Machine Intell.*, 11:325–329, 1989.

[115] M. H. Wright. Direct search methods: Once scorned, now respectable. In D. H. Griffiths and G. A. Watson, editors, *Numerical Analysis 1995*, pages 191–208. Addison-Wesley Longman, 1996.

[116] R. H. Zamar. Robust estimation in the errors-in-variables model. *Biometrika*, 76:149–160, 1989.

[117] Z. Zhang. Parameter-estimation techniques: A tutorial with application to conic fitting. *Image and Vision Computing*, 15:59–76, 1997.

[118] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *Intl. J. of Computer Vision*, 27:161–195, 1998.