Détection robuste des marquages routiers par une approche semi-quadratique

Robust Lane Marking Detection by the Half Quadratic Approach

Jean-Philippe Tarel, Sio-Song Ieng, Pierre Charbonnier

Octobre 2007/October 2007

Laboratoire Central des Ponts et Chaussées
58, bd Lefebvre, F 75732 Paris Cedex 15

Jean-Philippe TAREL
Chargé de Recherche
Division Exploitation, Signalisation, Éclairage
Laboratoire Central des Ponts et Chaussées

Sio-Song IENG
Ingénieur des Travaux Publics de l'État
ERA 17 LCPC
Laboratoire Régional des Ponts et Chaussées d'Angers

Pierre CHARBONNIER
Directeur de Recherche
ERA 27 LCPC
Laboratoire Régional des Ponts et Chaussées de Strasbourg

Ce document décrit les résultats obtenus par la collaboration entre la DESE (LCPC), le LIVIC (INRETS/LCPC), l'ERA 27 (LRPC de Strasbourg), et l'ERA 17 (LRPC d'Angers), entre 2002 et 2007, sur la perception par caméra des bords de voies routières

Pour commander cet ouvrage/*To order this book* :
Laboratoire central des ponts et chaussées
DISTC-section Diffusion
58 boulevard Lefebvre
F 75732 Paris Cedex 15
Tél : +33 1 40 43 50 20 - Fax : +33 1 40 43 54 95
Web page : http://www.lcpc.fr

Prix/*Price* :

En converture/*front cover* :
Exemples de résultats de détection des bords de voie sur différentes images de routes (images originales prisent par le LRPC Angers)
Examples of lane marking detections obtained on differents road images (original images grabbed by LRPC Angers)

Contents

Résumé **5**

Abstract **6**

Introduction **7**

1 Robust Curve Fitting in Images **9**
1.1 Introduction 9
1.2 Camera on board a vehicle 9
1.3 Perturbation Classification 9
1.4 Lane-Marking Extraction 11
1.4.1 Positive-Negative Gradients 11
1.4.2 Band Extraction 13
1.5 Road Shape Model 14
1.6 Parameters Estimation 16
1.6.1 Least Squares Fitting 16
1.6.2 Robust Parameters Estimation 17
1.6.3 Iterative Reweighted Least Square (IRLS) 18
1.7 Conclusion 19

2 Modeling Non-Gaussian Noise **21**
2.1 Introduction 21
2.2 Non-Gaussian Noise Models 22
2.3 Robustness Study 25
2.3.1 The SEF case 26
2.3.2 The GTF case 26
2.4 Noise Model Parameters Estimation 27
2.5 Sensitivity of Fitting to Noise Model Parameters 29
2.6 Improving Convergence: the GNC Strategy 31
2.7 Conclusion 35

3 Evaluating the Fits and Tracking **37**
3.1 Introduction 37
3.2 Exact Covariance Matrix for Least Squares Fitting 37
3.3 Approximate Confidence Matrices for Robust Fitting 38
3.3.1 Available Approximates 38

3.3.2 Confidence Matrix Comparison 39
3.3.3 Comparison on Synthetic Data 41
3.3.4 Correlated Data 42
3.4 Recursive Fitting 43
3.4.1 Recursive Least Square Fitting 44
3.4.2 Recursive Robust Fitting 45
3.5 Robust Kalman 46
3.6 Conclusion 49

4 Simultaneous Robust Fitting of Multiple Curves 51
4.1 Introduction 51
4.2 Multiple robust Maximum Likelihood Estimation (MLE) 51
4.3 Simultaneous Multiple Robust Fitting Algorithm 53
4.4 Connections with other approaches 54
4.5 Experimental Results 55
4.5.1 Geometric Priors 55
4.5.2 Lane-Markings Tracking 55
4.5.3 Camera Calibration 59
4.6 Conclusion 59

5 Conclusions 63

6 Appendix 65
6.1 Constrained Optimization in the IRLS 65
6.2 Covariance Matrix Derivation 66
6.3 Constrained Optimization in the SMRF 67

Robust Lane Marking Detection by the Half Quadratic Approach

Détection robuste des marquages routiers par une approche semi-quadratique

Jean-Philippe Tarel, Sio-Song Ieng, Pierre Charbonnier

Résumé

Détecter automatiquement les marquages horizontaux est un problème de base en analyse de scènes routières. Les applications concernent aussi bien l'inventaire des caractéristiques du marquage routier sur l'ensemble du réseau que la conception d'aides à la conduite embarquées sur véhicule. Pour mener à bien cette tâche, nous proposons dans ce document une modélisation tenant compte des variabilités géométriques du marquage et, surtout, robuste aux nombreuses perturbations observées dans les images relles. Le problème de la détection est alors formalisé comme un problème d'estimation, ce qui permet d'associer au résultat une mesure de confiance. Une telle auto-évaluation est, en effet, nécessaire pour intégrer la détection des marquages comme une brique de systèmes plus complexes, notamment dans les applications de suivi de voie. Nous présentons ici les algorithmes développés grâce à l'approche semi-quadratique de la théorie de la régression statistique robuste, que nous revisitons dans un formalisme lagrangien. L'approche développée permet une extension directe des algorithmes à la prise en compte simultanée de plusieurs lignes de marquage. Ces résultats ont été obtenus par une collaboration entre la DESE (LCPC), le LIVIC (INRETS/LCPC), l'ERA 27 (LRPC de Strasbourg), et l'ERA 17 (LRPC d'Angers), entre 2002 et 2007. Ils ont débouché sur un système de guidage opérationnel en temps réel, testé avec succès par le LIVIC dans le cadre du projet ARCOS 2004. D'autre part, un algorithme de détection de lignes multiples est utilisé en routine pour le calibrage géométrique des MLPC IRCAN (Imagerie Routière par CAméra Numérique).

Robust Lane Marking Detection by the Half Quadratic Approach

Détection robuste des marquages routiers par une approche semi-quadratique

Jean-Philippe Tarel, Sio-Song Ieng, Pierre Charbonnier

Abstract

Automatic road marking detection is a key point in road scene analysis, whose applications concern as well the inventory of lane marking on the road network as the design of assistance systems on-board vehicles. To this end, we propose in this document a model which accounts for the geometric variability of the markings and, above all, which is robust to the numerous perturbations that can be observed in real-world images. The detection problem is formalized as an estimation problem. This allows to associate each result with a confidence measure. Such a self-evaluation capacity is indeed necessary when integrating road marking detection into more complex systems, such as lane following applications. We present here algorithms that can be derived thanks to the half-quadratic approach of statistical robust estimation, which we revisit in a Lagrangian formalism. The proposed approach allows a straightforward extension of the estimation algorithms to the simultaneous fitting of multiple marking lines. These results were obtained in collaboration with DESE (LCPC), LIVIC (INRETS/LCPC), ERA 27 (LRPC Strasbourg), and ERA 17 (LRPC Angers), between 2002 and 2007. They lead to an operational, real-time guidance system, which was successfully tested within the context of the ARCOS 2004 project. Moreover, a multiple lines detection algorithm is routinely used for calibrating the MLPC IRCAN (French acronym for Road Imagery using Numeric CAmeras) systems.

Introduction

Accurate and robust road marking detection is a key point in road scene analysis, whose applications concern both road management and the automotive industry. Indeed, measuring road marking features is a basic task for road inventory systems. Moreover, localizing a vehicle with respect to the lane marking is of main importance for driver assistance systems.

In this kind of *computer vision* applications, cameras and their associated image analysis algorithms are increasingly being considered as specialized sensors. As for every sensor, the task of the vision device requires robustness to perturbations, especially when safety aspects are involved. Moreover, each of its outputs must be accompanied by some *confidence* measure in order to be integrated into complex control systems. In this work, we address the problem of designing such a robust and self-evaluating system.

To this end, as in many image analysis applications, an efficient solution is to formalize detection as a *fitting* problem, as explained in Chapter 1. More precisely, our approach involves a *marking extractor* algorithm, whose outputs, namely putative road marking centers, are approximated by curves. We propose several curve models to handle the geometric variability of lane markings observed under different camera configurations. While the curves are, in general polynomials, the model itself is linear with respect to (w.r.t.) its parameters, which contributes to the simplicity of the formalism. This linear, deterministic *generative model* is associated with a noise model that accounts for the perturbations of the data, or noise. In this framework, fitting simply amounts to estimating the parameters of the generative model under the noise assumption. For example, considering Gaussian noise leads to least-squares fitting, which is well known, widely used, and provides the exact covariance matrix of the obtained fit as a natural confidence measure. However, it is common knowledge that in real applications, image perturbations are seldom Gaussian, which makes the fitting task more difficult. The ultimate reason is that Gaussian models assign very small probability to high *residuals* corresponding to data that lay far from the model, or *outliers*. In the robust statistical estimation framework, it is possible to deal with *heavy tailed* noise models. Unfortunately, this leads to solving systems of non-linear equations. However, iterative algorithms such as Iteratively Reweighted Least Squares (IRLS) [20] were devised to circumvent this difficulty and obtain a robust fit in the context of *M-estimators* [20]. The class of potential functions for which the convergence of the IRLS algorithm is proved has been extended w.r.t. the original M-estimation theory by the half-quadratic framework [14, 6], which we propose to revisit here in a Lagrangian formalism.

In robust estimation, the choice of the potential function or, equivalently, of the noise model, is an important issue, which is seldom discussed in the literature. In Chapter 2, we analyze this problem, which may well be encountered in other robust detection or tracking algorithms. This leads us to introduce two families of noise models that allow continuous transition between Gaussian and non Gaussian models, with only two parameters. This property can be used

to better approximate the real noise distribution and thus, to improve the estimation accuracy of the algorithm. We propose a study of the two introduced families in terms of robustness, following the lines of Mizera and Müller's work [30]. Finally, the choice of the potential may have important consequences in terms of the sensitivity of the fitting result to the tuning of the parameters, especially when using very non-convex potentials. We also address this question and propose methods for automatically estimating the parameters.

Chapter 3 is devoted to the presentation of the way the output of the algorithm is evaluated. The question of devising a confidence measure in the context of M-estimators is also seldom addressed in the computer vision literature. Due to the non-linearities, an exact definition of the covariance matrix seems intractable and approximations are required. One way to tackle the problem is to study the asymptotic behavior of the robust estimator: Huber [20] proposes several approximate covariance matrices. To our knowledge, the evaluation of these matrices has only been performed on synthetic data with large data sets. In a different context, namely robust Kalman filtering, where the question of the covariance matrix becomes of major importance, other approximates were proposed in [8]. However, no theoretical justification was proposed and the validation was only performed on simulations. We present an experimental study that we carried out on real data to compare several covariance matrix approximates, already proposed in the literature, under strong perturbations. We introduce a new approximate covariance matrix, where noise correlation is taken into account, which is tested under different experimental conditions. Note that, unlike those proposed by Huber, it is not an asymptotic covariance matrix. To close this chapter, the way covariance matrices can be used in recursive algorithms for tracking along an image sequence is explained.

Finally, in Chapter 4, we illustrate the power of our Lagrangian approach of robust estimation, by presenting its application to the simultaneous fitting of multiple curves. An iterative algorithm, which extends the IRLS to simultaneous multiple fitting, is readily obtained. The proposed algorithm simultaneously performs features clustering and fitting. It bears certain similarities with existing methods such as EM (Expectation-Maximization, [11]), but is derived in a more straightforward way. Applications to lane tracking and to camera calibration are finally presented.

Chapter 1

Robust Curve Fitting in Images

1.1 Introduction

In this chapter, the application constraints are first described and, second, we justify our approach to lane-marking detection, in Sections 1.2 and 1.3. Lane-marking detection is performed in two steps: a step of marking features extraction detailed in Section 1.4, followed by robust curve fitting. The last step is based on the use of a specific road shape model described in Section 1.5. The robust curve fitting algorithm is derived using Lagrange formalism in Section 1.6. Most of the material of this chapter was first publish in [40].

1.2 Camera on board a vehicle

The goal of using sensors in road inventory and driving assistance systems is to provide reliable and accurate information about the road and the vehicle's environment. One important example of such an application is lane-marking detection and tracking which must provide, in real-time, lateral distance to lane-markings, road curvature, and lane-marking features.

The kind of parameter that can be estimated by the vision system depends on the camera's orientation and position. For instance, using a side-viewing camera, the relative location of the vehicle w.r.t the center of the road can be accurately estimated. When the camera is looking forward, the lateral position of the vehicle may be less accurate, but the curvature and more generally the path geometry in front of the vehicle can be estimated. This is why most vision systems for lane-markings detection use a frontal camera.

In the following, the proposed algorithms applies to several cameras that can be used and mounted at different positions (frontal and/or lateral) to better estimate the path geometry as well as the relative position of the vehicle (see Figure 1.1) as described in [25].

1.3 Perturbation Classification

One advantage of the camera as a sensor is that a great deal of geometric and photometric information about the scene can be acquired. But this can also cause difficulties: the useful information may be drowned in clutter due to trees, advertisement panels or clouds, for instance.

Figure 1.1: *Top line: camera looking forward on the LIVIC's prototype vehicle and a typical image seen from the frontal camera.*
Bottom line: side-viewing camera on the prototype vehicle and an image taken by this camera. The pixel coordinate system (x, y) *is shown in the top-right image*

Note that some of this perturbation can be removed by extracting a set of properly chosen features from the image, in a pre-detection step we name *extraction*. On the other hand, another possible source of difficulty that may strongly affect the detection process is a lack of information. In our application, this is usually due to erased parts of lane-markings, or occlusions by obstacles. Finally, we can classify perturbations into two main classes:

1. noise, including sensor noise and clutter noise (*outliers),*
2. lack of information: erased markings or occlusions, adverse illumination conditions.

How does the proposed approach deal with these problems? The first class of perturbations is handled by the two step architecture of our detector: a *conservative* lane-marking feature extractor followed by a robust curve fitting algorithm. To deal with the second class of perturbation, some kind of temporal interpolation can be implemented by embedding the detector in a Kalman filter, as explained in Chapter 3.

1.4 Lane-Marking Extraction

In this Section, we propose two extraction algorithms for lane-marking extraction. A classical edge detector provides too much information as far as lane-marking detection alone is concerned. To our knowledge, only a few low-level algorithms use geometric criteria to select the interesting edges. However, accounting for geometrical criteria in addition to photometric ones proves to be a very efficient strategy [16, 36]. In the case of road markings, for example, their constant width provides a useful selection criterion.

1.4.1 Positive-Negative Gradients

The first proposed extractor processes every line of the image in a sequential fashion. It first selects intensity gradients with a norm greater than a given threshold, S_G. Since our aim is to avoid missing any feature, even in adverse lighting conditions, we set S_G as small as possible. This is the so-called *conservative assumption*, also used in [16, 13], for instance.

Then, the algorithm selects a pair of positive and negative gradients according to their separating distance. To our knowledge, only a few low-level algorithms take geometric criteria into account in addition to photometric criteria. However, it proves to be a very efficient strategy [16], especially when man-made, standardized features such as lane markings are considered. For each image line x, let y_{init} be the first position for which a horizontal gradient is greater than the threshold S_G. Let y_{end} be the pixel position where the image intensity is then strongly decreasing (the coordinates, x and y, being defined as shown in Figure 1.1). A lane-marking feature is extracted if y_{end} - y_{init} lies within the range $[S_m, S_M]$. The limits are supposed to follow a linear model: $S_m = C_m x + D_m$ and $S_M = C_M x + D_M$. This model is valid both for side-viewing cameras (with $C_m = C_M = 0$) as well as for front-viewing cameras, as it is known that in the latter case, the observed lane-marking width decreases linearly and reaches zero at the horizon line. In a classical single-camera model, we have for C_m, C_M, D_m and D_M:

$$C_i = \frac{l_i \alpha_y cos(\theta)}{\alpha_x h} \qquad D_i = -\frac{l_i \alpha_y cos(\theta) x_0}{\alpha_x h} - \frac{l_i \alpha_y sin(\theta)}{h}, \tag{1.1}$$

where θ is the angle of the camera axis w.r.t the horizon, h is the height of the camera, α_y and α_x are respectively the inverse of pixel width and length, (y_0, x_0) is the center of the image in pixels, l_m and l_M are the minimal and maximal marking widths on the road.

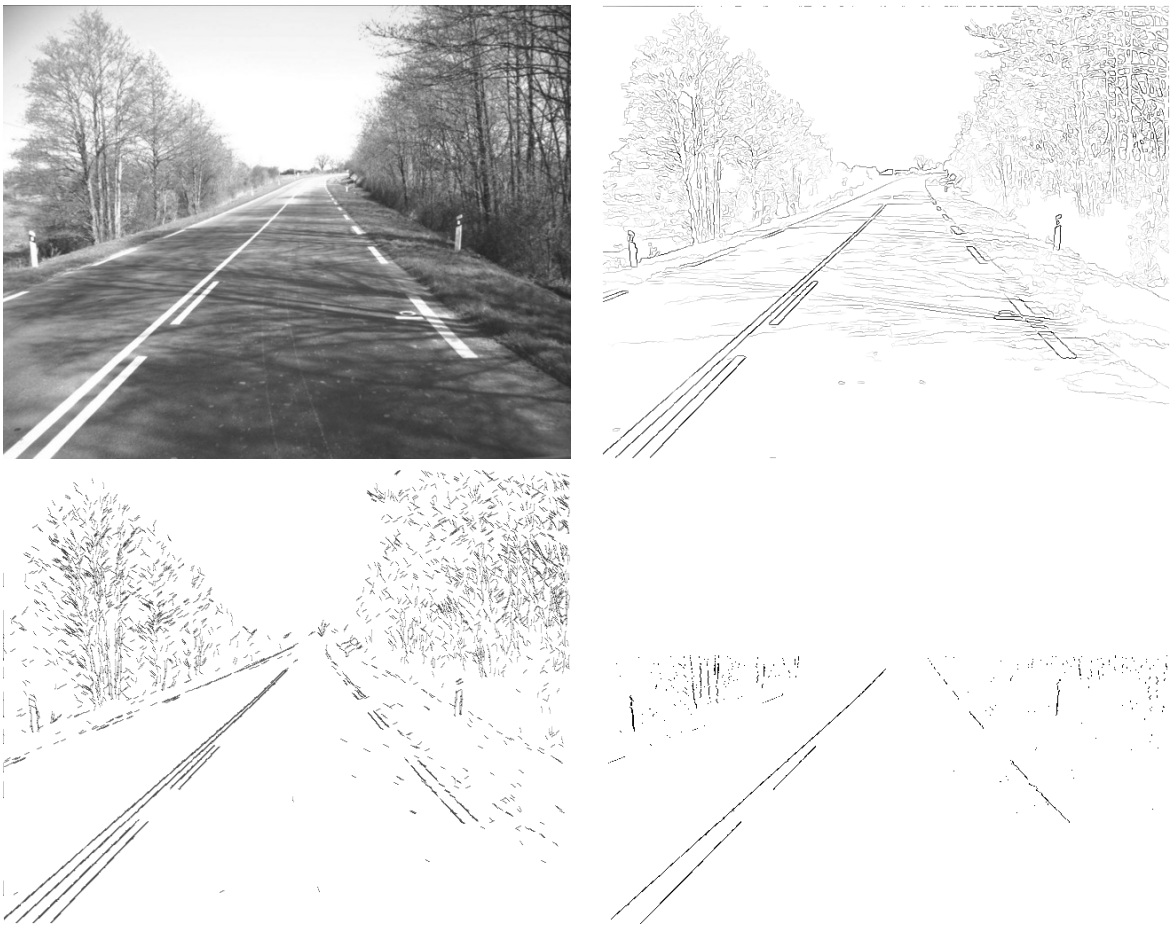


Figure 1.2: *Top left: Original image of a road.*
Top right: result of Canny's filter.
Bottom left: result of Segment extraction [16].
Bottom right: result with the proposed lane-marking feature extractor (in this case, only the medial axis of each detected region is shown)

Finally, the centers of the selected couples of edges are used as features. The algorithm for every image line is:

- $y = 1$
- `While` $y < ImageLineSize$,

  1. `Compute the horizontal gradient` $G(y)$
  2. `If` $G(y) > S_G$ `then,`
     - $y_{init} = y$

- $I = Intensity(y_{init}) + \frac{G(y_{init})}{2}$
- $y = y_{init} + 1$
- `While` $Intensity(y) > I$ `and` $y < ImageLineSize$,
  - $y = y + 1$
- $y_{end} = y$
- `If` $y_{end} - y_{init} \in [S_m, S_M]$ `then`
  - `A lane-marking feature is extracted with its center at` $\frac{(y_{end}+y_{init})}{2}$
  - $y = y_{end} + 1$
- `else,` $y = y_{init} + 1$

3. `else,` $y = y + 1$.

Figure 1.2 shows the superiority of the proposed extractor w.r.t classical edge detectors in an adverse situation. The proposed extractor only yields a small number of outliers despite shadows and highlighted areas. Moreover the lane-markings are completely extracted. Note that extracting the center of the road marking rather than its edges avoids selecting redundant information that would be useless for the next processing steps.

1.4.2 Band Extraction

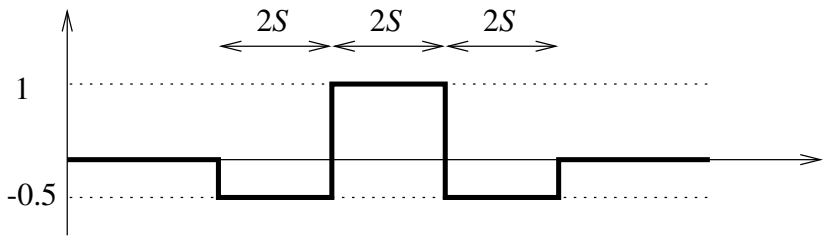


Figure 1.3: *Band filter for marking extraction*

When a road marking is dirty or worn-out, the condition $Intensity(y) > I$ may not always be verified along its whole surface, and the previous extractor may not be able to completely extract it. This is why, following Thomas Veit's (LIVIC) suggestions, we propose a new extractor based on the line convolution with the band filter shown in Fig. 1.3. As previously, the extractor processes every line of the image in a sequential fashion. Note that the convolution with the filter can be performed in only 4 operations: $\frac{1}{4S}(3(C(y+S) - C(y-S)) - (C(y+3S) - C(y-3S)))$, independently of its width $6S$, by using the cumulative function $C(y)$ of the intensity profile along the line. The convolution is performed with several widths and the obtained results are summed. Finally, the local maxima of the summed convolutions are considered as putative marking centers and used as features. The algorithm for every image line is:

- $C(0) = 0$
- `For` $y = 1$ `to` $ImageLineSize$,

  - $C(y) = C(y-1) + Intensity(y-1)$,
  - $convsum(y) = 0$

- `For` $S = S_m/2$ `to` $S_M/2$,
  - `For` $y = 3S$ `to` $ImageLineSize - 3S$,
    * $convsum(y) += \frac{1}{4S}(3(C(y+S) - C(y-S)) - (C(y+3S) - C(y-3S)))$

- `For` $y = 3S$ `to` $ImageLineSize - 3S$,
  - `if` $(convsum(y) \geq convsum(y+1))$ `and` $(convsum(y) \geq convsum(y-1))$ `a lane-marking feature is extracted with its center at` y.

In figure 1.4, a comparison of the positive-negative gradients and band filter extractors is shown. The same parameters are used by the two extractors. On the first row, two original images are displayed and in the two following rows, areas extracted as marking appears in blue. A search for the best S given a center position was thus added after filter band extraction. In the last row, Receiver Operating Characteristic (ROC) curves are given. ROC curves display the false alarm rate versus the correct detection rate and allows to compare extractor performances independently of the value of the threshold parameter. While the positive-negative gradients extractor does not extract highlighted areas, it is not always able to extract dirty or erased markings. On the contrary, the band filter extractor performs better on dirty or erased markings, but highlighted areas are also extracted as markings. As a consequence, to follow the conservative assumption, *i.e.*, to avoid missing any feature, the band filter must be preferred.

1.5 Road Shape Model

In this section, we describe the curve model we use for aggregating the detected features into marking lines. This model is more detailed in [36].

In many road inventory and driving assistance systems, the marking detection has to be performed over a distance of 10 to 40m. Therefore, a flat road model, which neglects the vertical curvature of the road, seems to be a reasonable approximation. Its main advantage is that it allows the road shape to be estimated from only one camera. The road shape features are given by the lane-marking centers extracted as explained in Section 1.4. We assume that the extracted features are noisy measurements of a curve explicitly described by a linear parametric model:

$$\tilde{y} = \sum_{i=0}^{d} f_i(x)\tilde{a}_i = X(x)^t \tilde{A}, \tag{1.2}$$

where $(x, \tilde{y})$ are the image coordinates of a point on the curve, $\tilde{A} = (\tilde{a}_i)_{0 \leq i \leq d}$ is the coefficient vector of the curve parameters and $X(x) = (f_i(x))_{0 \leq i \leq d}$ is the vector of basis functions of the image coordinate x.

The linearity of this generative model (1.2) is particularly useful in setting up the shape estimation problem as a fitting problem and thus, in speeding up the detection. In practice, we model lane-markings on the road by polynomials $y = \sum_{i=0}^{d} a_i x^i$. While the clothoid model would certainly be more realistic for side-viewing cameras, it would also be much more complex to fit.

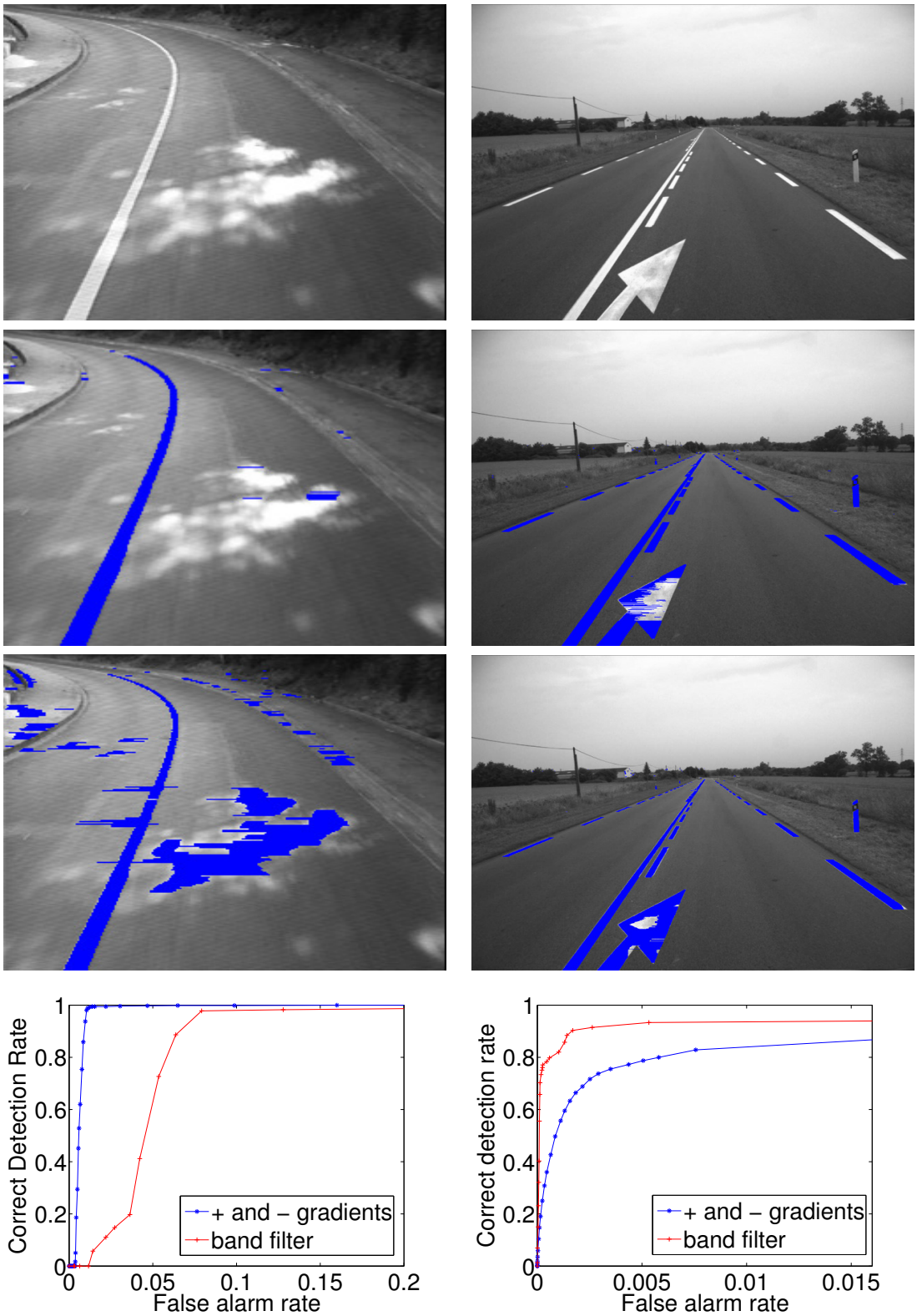


Figure 1.4: *First row: original images.*
Second row: results of positive-negative gradients extractor.
Third row: band filter extractor.
Last row: Receiver Operating Characteristic (ROC) curves of the two extractors

Fortunately, it can be well approximated with a polynomial as proposed in [12]. Moreover, the image of a polynomial on the (flat) road under perspective projection is a *hyperbolic polynomial* with equation $y = c_0 x + c_1 + \sum_{i=2}^{d} \frac{c_i}{(x-x_h)^i}$, where c_i is linearly related to a_i as detailed in [39, 36]. Therefore, a hyperbolic polynomial model can also be used for a frontal camera. In this case, however, hyperbolic polynomials may advantageously be replaced by the so-called *fractional polynomials*, defined by $y = \sum_{i=0}^{d} d_i x^{\frac{i}{d}}$, for the sake of numerical stability.

To avoid numerical problems, a whitening of the data is performed by scaling the image in a $[-1,1] \times [-1,1]$ box for polynomial curves and in a $[0,1] \times [-1,1]$ box for fractional polynomials, prior to the fitting.

In model (1.2), the vertical coordinate x is assumed non-random. Thus only the second coordinate of the extracted point, y, is considered as a noisy measurement:

$$y = X(x)^t \tilde{A} + b \tag{1.3}$$

In all that follows, the measurement noise b is assumed independent and identically distributed (iid), and centered. In the next section, we explain how the road shape model can be robustly fitted for lane-marking detection in images.

1.6 Parameters Estimation

In this section, we first review Least Squares fitting, then our approach to robust fitting is detailed within a Lagrangian formalism.

1.6.1 Least Squares Fitting

First, we recall the very simple situation where the noise b is Gaussian. The goal is to estimate the curve parameters A_{LS} on the whole set of n extracted points (x_i, y_i), $i = 1, ..., n$. Let $\tilde{A}$ denote the vector of "true" underlying curve parameters. Let σ be the standard deviation of the Gaussian noise b. The probability of a measurement point (x_i, y_i), given the curve parameters A, or likelihood, is:

$$p_i((x_i, y_i)/A) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}(\frac{X_i^t A - y_i}{\sigma})^2}.$$

For the sake of simplicity, from now on, we note $X_i = X(x_i)$. We can write the probability of the whole set of points as the product of the individual probabilities:

$$p \propto \frac{1}{\sigma^n} \prod_{i=1}^{i=n} e^{-\frac{1}{2}(\frac{X_i^t A - y_i}{\sigma})^2}, \tag{1.4}$$

where $\propto$ denotes the equality up to a factor. Maximizing the likelihood, p, is equivalent to minimizing the negative of its logarithm w.r.t A. In the Gaussian case, this leads to minimizing the so-called Least Squares error:

$$e_{LS}(A) = \frac{1}{2\sigma 2} \sum_{i=1}^{i=n} (X_i^t A - y_i)^2.$$

The minimization of e_{LS} is equivalent to solving the well-known normal equations:

$$XX^tA = XY, \tag{1.5}$$

where $Y = (y_i)_{1\leq i\leq n}$ is the vector of y coordinates, the matrix $X = (X_i)_{1\leq i\leq n}$ is the *design matrix*, and $S = XX^t$ is always symmetric and positive. If S is not singular, (1.5) admits the unique solution $A_{LS} = S^{-1}XY$. Computing the best fit A_{LS} simply requires solving the linear system (1.5). As just explained, it is also the Maximum Likelihood Estimate (MLE) assuming Gaussian noise.

Since Y only is random, the expectation of A_{LS} is $E[A_{LS}] = S^{-1}XE[Y]$. The point coordinates in $E[Y]$ correspond to points exactly on the underlying curve $\tilde{Y}$, thus $\tilde{A} = S^{-1}X\tilde{Y}$. Therefore, $E[A_{LS}]$ equals $\tilde{A}$, *i.e.* the estimator A_{LS} of $\tilde{A}$ is unbiased.

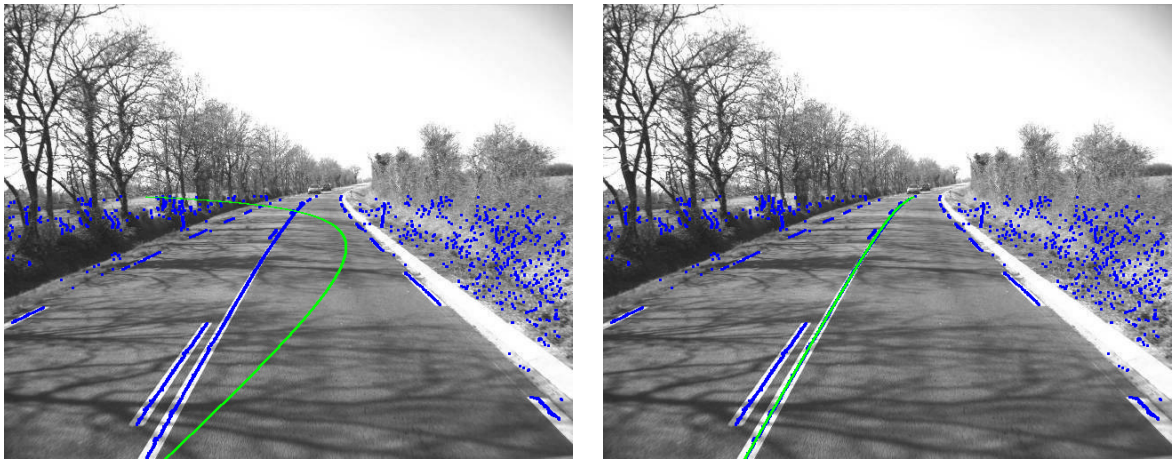


Figure 1.5: *Road marking features (in blue) and estimated lane-marking (in green) assuming Gaussian noise (left image) and T-Student noise, as defined in Sec. 2.2 (right image)*

As is well known (see Figure 1.5), Least Squares fitting does not provide correct fits in the presence of image perturbations.

1.6.2 Robust Parameters Estimation

We still assume that the noise is iid and centered. But now, the noise is assumed to be non Gaussian. In particular, it may exhibit heavier tails than the Gaussian probability distribution function (pdf). It can be chosen in one of the families that will be presented in Chapter 2. The assumed noise model is specified by a function $\phi(t)$ in such a way that the probability of measurement point (x_i, y_i), given curve parameter A, is:

$$p_i((x_i, y_i)/A) \propto \frac{1}{s} e^{-\frac{1}{2}\phi((\frac{X_i^t A - y_i}{s})^2)} \tag{1.6}$$

In this Section, we revisit the derivation of the estimate using a Lagrangian approach which leads to the same algorithm as that obtained by the half-quadratic or M-estimation approach. The advantage of the Lagrangian framework is to allow a simple proof of local convergence and an interpretation in terms of dual optimization. As in the half-quadratic approach [14, 6], $\phi(t)$ satisfies the following hypotheses:

- **H0**: defined and continuous on $[0,+\infty[$ as its first and second derivatives,
- **H1**: $\phi'(t) > 0$ (thus ϕ is increasing),
- **H2**: $\phi''(t) < 0$ (thus ϕ is concave).

These three assumptions are very different from those used in the M-estimation approach for the convergence proof. Indeed in [20], the convergence proof requires that potential function $\rho(u) = \phi(u^2)$ be convex. In our case, the concavity and monotony of $\phi(t)$ implies that $\phi'(t)$ is bounded, but potential $\phi(u^2)$ is not *necessarily convex* w.r.t. u. Note that the smooth exponential family (for $\alpha < 1$) and the generalized T-Student family (for $0 < \beta$) that will be presented in Chapter 2, both satisfy these three assumptions.

As stated in [20], the role of this function ϕ is to saturate the error in case of a large scaled residual $|b_i| = |X_i^t A - y_i|$, and thus to lower the importance of outliers. The scale parameter, s, sets the residual value from which a noisy data point has a good chance of being considered as an outlier.

In the MLE framework, the problem is set as the minimization w.r.t. A of the robust error:

$$e_R(A) = \frac{1}{2}\sum_{i=1}^{i=n} \phi((\frac{X_i^t A - y_i}{s})^2). \tag{1.7}$$

Notice that while the Gaussian case would correspond to the particular case in which $\phi(t) = t$, this function does satisfy assumption **H2**. $e_{LS}(A)$ is indeed a limit case of $e_R(A)$. Unlike in the Gaussian case, the minimization problem is not quadratic. It could be solved iteratively using Gradient or Steepest Descent algorithms. However, these algorithms can be relatively slow around local minima, when the gradient slope is near zero. Indeed, their speed of convergence is only linear, while Newton algorithms achieve a quadratic speed of convergence. But generally, with Newton algorithms, the convergence to a local minimum is not certain. However, we prove that the Newtonian construction presented in the next paragraph always converges toward a local minimum. A global minimum could be obtained using simulated annealing, but at a much greater computational cost.

In the two following sections, we present the algorithm derived from (1.7) within Lagrange's formalism. The approach consists in first rewriting the problem as the search for a saddle point of the associated Lagrange function. Then, the algorithm is obtained by alternated minimizations of the dual function. As we will see, this algorithm corresponds to the well known algorithm named Iterative Reweighted Least Squares (IRLS).

1.6.3 Iterative Reweighted Least Square (IRLS)

First, we rewrite the minimization of $e_R(A)$ as the maximization of $-e_R$. This will allow us later to write $-e_R(A)$ as the extremum of a convex function rather than a concave one, since the negative of a concave function is convex. Second, we introduce the auxiliary variables $w_i = (\frac{X_i^t A - y_i}{s})^2$. These variables are needed to rewrite $-e_R(A)$ as the value achieved at the minimum of a constrained problem. This apparent complication is in fact valuable since it allows us to introduce the Lagrange multipliers. Indeed using **H1**, $-e_R(A)$ can be seen as the

minimization w.r.t. $W = (w_i)_{1 \leq i \leq n}$ of:

$$E(A,W) = \frac{1}{2}\sum_{i=1}^{i=n} -\phi(w_i),$$

subject to n constraints $h_i(A,W) = w_i - (\frac{X_i^t A - y_i}{s})^2 \leq 0$. The following algorithm is derived from this constrained optimization problem as shown in appendix 6.1. The IRLS algorithm is:

1. Initialize A_0, and set $j = 1$,
2. For all indexes i ($1 \leq i \leq n$), compute the auxiliary variables $w_{i,j} = (\frac{X_i^t A_{j-1} - y_i}{s})^2$ and the weights $\lambda_{i,j} = \phi'(w_{i,j})$,
3. Solve the linear system $\sum_{i=1}^{i=n} \lambda_{i,j} X_i X_i^t A_j = \sum_{i=1}^{i=n} \lambda_{i,j} X_i y_i$,
4. If $\|A_j - A_{j-1}\| > \varepsilon$, increment j, and go to 2, else $A_{RLS} = A_j$.

The convergence test can also be performed on the error variation. A test on a maximum number of iterations can also be added.

Finally, Figure 1.6 illustrates the importance of robust fitting in images with many outliers. The yellow lines depict the initial A_0's. The green ones are the fitting results A_R, assuming Gauss, smooth Laplace, Cauchy, and Geman & McClure [15] noises (as defined in the next chapter). A correct fit is achieved only with the last two pdfs, which correspond to non-convex potentials. This illustrates the importance of the choice of a noise model, as detailed in the next chapter.

1.7 Conclusion

We described the design of a camera-based lane-marking detection system involving a feature extraction step followed by a robust fitting step. The feature extraction takes into account the lane-marking geometry, particularly its width. This criterion allows us to better select features while following the conservative assumption, which means that outliers are removed even if very noisy data must be kept. These features are robustly fitted by a curve, using the well-known Iterative Reweighted Least Squares algorithm. We have shown how Lagrange multipliers lead to a revised half-quadratic theory and justify the local convergence of the algorithm even for non-convex potential, unlike M-estimators. Indeed, the half-quadratic framework is valid for a large class of non-convex potentials which leads to *more robust* estimators.

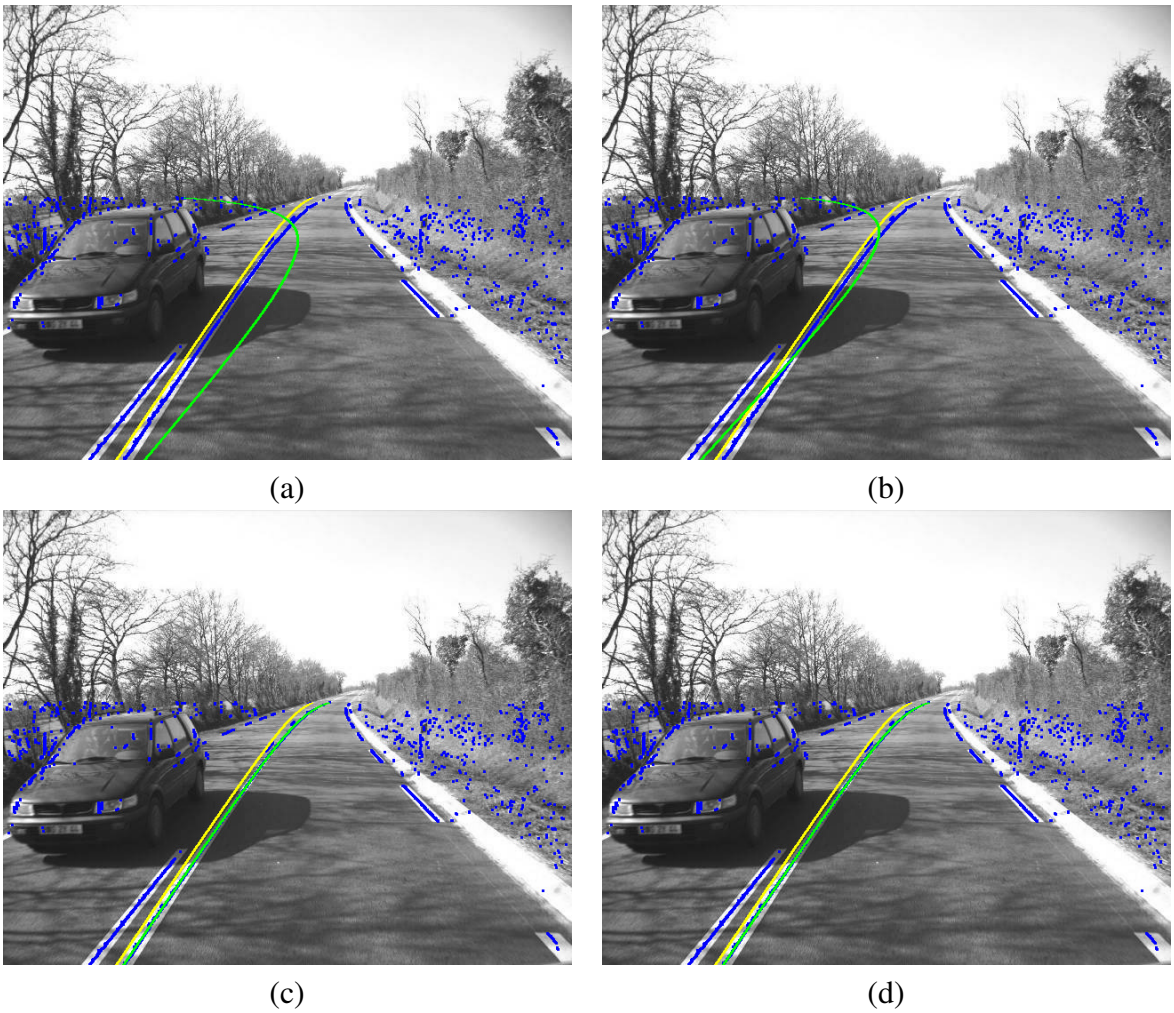


Figure 1.6: *Fitting on a real image assuming Gauss (a), smooth Laplace (b), Cauchy (c), and Geman & McClure (d) noise pdfs. Data points are shown in blue, yellow lines are the initial* A_0*'s and green lines are the fitting results. The noise models used in this experiment are defined in Chapter 2*

Chapter 2

Modeling Non-Gaussian Noise

2.1 Introduction

In computer vision, data are most of the time corrupted by non-Gaussian noise or outliers and may contain multiple statistical populations. It is a difficult task to model observed perturbations. Several parametric models were proposed in [19, 35], and sometimes based on mixtures [18]. Non-Gaussian noise models imply using robust algorithms to reject *outliers*. The most popular techniques are Least Median Squares (LMedS), RANSAC and Iterative Reweighted Least Squares (IRLS). The first two algorithms are close in their principle and achieve the highest *breakdown point*, *i.e.* the admissible fraction of outliers in the data set, of approximately 50%. However, their computational demand quickly increases with the number of parameters. We focus on the IRLS algorithm and one may argue that it is a deterministic algorithm that only converges towards a local minimum close to its starting point, as explain in the previous chapter. This difficulty can be circumvented by the so-called Graduated Non Convexity (GNC) strategy, that will be presented in this chapter. The IRLS algorithm, even within the GNC strategy, usually remains faster compared to LMedS and RANSAC and is also able to achieve the highest breakdown point.

We now rewrite (1.3), and thus let us consider the linear problem:

$$Y = XA + B \tag{2.1}$$

where $Y = (y_1, \cdots, y_n) \in \mathbb{R}^n$ is a vector of observations, $X = (\mathbf{x}_1, \cdots, \mathbf{x}_n) \in \mathbb{R}^{n \times d+1}$ the design matrix, $A \in \mathbb{R}^{d+1}$ the vector of unknown parameters that will be estimated by the IRLS algorithm, and $B = (b_1, \cdots, b_n) \in \mathbb{R}^n$ the noise. The noise is assumed independent and identically distributed but not necessarily Gaussian. We consider the *fixed design* setting, *i.e.* in (2.1), X is assumed non-random. In that case, as demonstrated in [30], certain M-estimators achieve the maximum possible breakdown point. However, if X cannot be assumed non-random, the breakdown point of M-estimators drops towards zero [28]. This underlines the importance of the way computer vision problems are formulated.

This chapter is organized as follows. In Section 2.2, we present the non-Gaussian noise models we found of interest for lane-marking detection. Then in Section 2.3, we prove that M-estimators that achieve the maximum breakdown point of approximately 50% can be built based on these probability distribution families. This choice may have important consequences in terms of the sensitivity of the fitting result to the tuning of the parameters, especially when it

leads to a non-convex minimization. In Section 2.4 and Section 2.5, we also address this question and propose methods for automatically estimating these parameters. Finally, we describe advantages of so-called Graduated Non Convexity (GNC) strategy in Section 2.6. Most of this chapter was first published in [22, 26, 24].

2.2 Non-Gaussian Noise Models

We are interested in parametric functions families that allow a continuous transition between different kinds of probability distributions.

We here focus on two simple parametric probability density functions (pdf) of the form $pdf(b) \propto e^{-\rho(b)}$ suitable for the IRLS algorithm, where $\propto$ denotes the equality up to a factor. Compared to the previous chapter, where the pdf was written as a function of ϕ, see (1.6), we use in this chapter mainly the ρ notation, for a better compatibility with Mizera and Müller's work [30].

A first interesting family of pdfs is the stretched exponential family (also called generalized Laplacian, or generalized Gaussian [35]):

$$E_{\alpha,s}(b) = \frac{\alpha}{s\Gamma(\frac{1}{2\alpha})} e^{-((\frac{b}{s})^2)^\alpha} \tag{2.2}$$

The two parameters of this family are the scale s and the power α. The latter specifies the shape of the noise model. Moreover, α allows a continuous transition between two well-known statistical laws: Gaussian ($\alpha = 1$) and Laplacian ($\alpha = \frac{1}{2}$). The associated ρ function is $\rho_{E_\alpha}(\frac{b}{s}) = ((\frac{b}{s})^2)^\alpha$. The associated ϕ function is $\phi_{E_\alpha}(t) = t^\alpha$, with $t = \frac{b^2}{s^2}$.

As detailed in Chapter 1, to guarantee the convergence of IRLS, the ϕ' function, related to ρ' by $\phi'(u^2) = \frac{\rho'(u)}{2u}$ with $u = \frac{b}{s}$, has to be defined on $[0, +\infty[$. This is not the case for $\alpha \leq 0$ in the stretched exponential family. Therefore, the so-called smooth exponential family (SEF) $S_{\alpha,s}$ was introduced in [40]:

$$S_{\alpha,s}(b) \propto \frac{1}{s} e^{-\frac{1}{2}\rho_\alpha(\frac{b}{s})} \tag{2.3}$$

where $\rho_\alpha(u) = \frac{1}{\alpha}((1+u^2)^\alpha - 1)$. The associated ϕ function is $\phi_{S_\alpha}(t) = \frac{1}{\alpha}((1+t)^\alpha - 1)$.

Table 2.1: *list of noise models used in robust fitting (SEF)*

α	$\phi_{S_\alpha}(t)$	weight=$\phi'_{S_\alpha}(t)$	pdf name
1	t	1	Gauss
$\frac{1}{2}$	$2(\sqrt{1+t}-1)$	$\frac{1}{\sqrt{1+t}}$	smooth Laplace
0	(not defined)	$\frac{1}{1+t}$	generalized T-Student
-1	$\frac{t}{(1+t)}$	$\frac{1}{(1+t)^2}$	Geman & McClure

As for the stretched exponential family, α allows a continuous transition between well-known statistical laws such as Gauss ($\alpha = 1$), smooth Laplace ($\alpha = \frac{1}{2}$) and Geman & McClure ($\alpha = -1$), see Tab. 2.1. These laws are shown on Figure 2.1. Note that, for $\alpha \leq 0$, $S_{\alpha,s}$ can

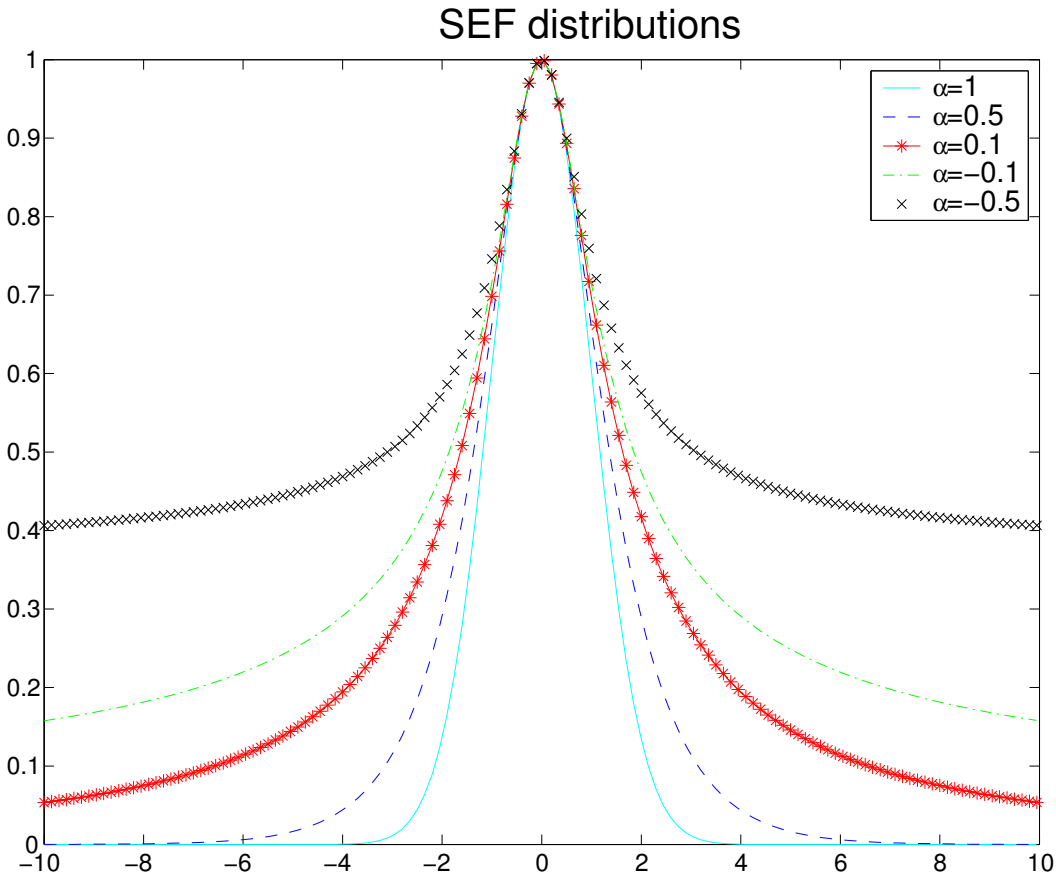


Figure 2.1: *SEF noise models, $S_{\alpha,s}$. Notice how tails are getting heavier as α decreases*

be always normalized on a bounded support, so it can still be seen as a pdf. In the smooth exponential family, when α is decreasing, the probability to observe large, not to say very large errors (outliers), increases.

In the IRLS algorithm, the residual b is weighted by $\lambda = \phi'(\frac{b^2}{s^2})$. For the SEF, the associated weight is $\phi'_{S_\alpha} = (1+t)^{\alpha-1}$, see Tab. 2.1. Notice that while the pdf is not defined when $\alpha = 0$, its weight does and it corresponds in fact to the T-Student law. We define the Generalized T-Student Family (GTF) by:

$$T_{\beta,s}(b) = \frac{\Gamma(-\beta)}{\sqrt{\pi}\Gamma(-\beta-\frac{1}{2})s}(1+\frac{b^2}{s^2})^{\beta} \tag{2.4}$$

where $\beta < 0$. This family of pdfs also satisfies the required properties for robust fitting. It is named generalized T-Student pdf [19] in the sense that an additional scale parameter is introduced compared to the standard T-Student pdf. Notice, that the case $\beta = -1$ corresponds to the Cauchy pdf. These laws are shown in Figure 2.2. The GTF can be rewritten as:

$$T_{\beta,s}(b) \propto \frac{1}{s}e^{-\frac{1}{2}\rho_\beta(b)} \tag{2.5}$$

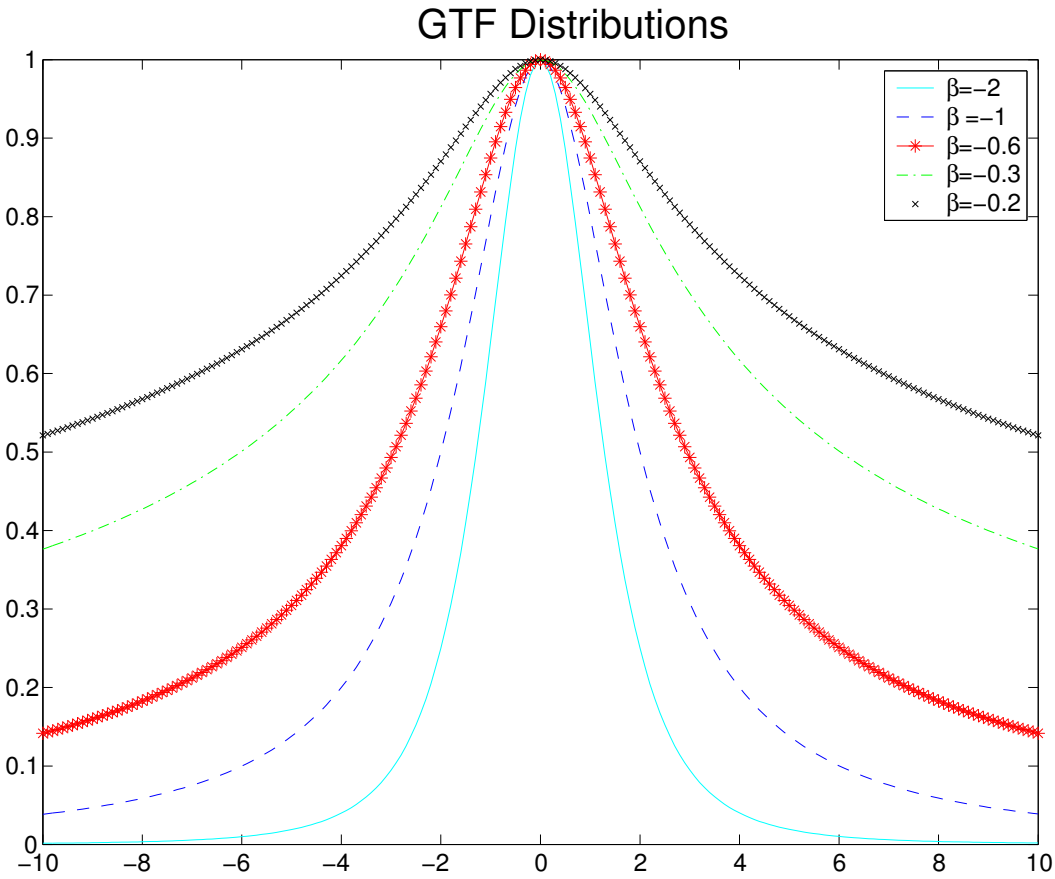


Figure 2.2: *GTF noise models, $T_{\beta,s}$. Notice how tails are getting heavier when β is increasing towards* 0

where $\rho_\beta(\frac{b}{s}) = -2\beta\ln(1+\frac{b^2}{s^2})$. The associated ϕ function is $\phi_{T_\beta}(t) = 2\beta ln(1+t)$. Moreover, it is easy to show that the so-called generalized T-Student pdfs have the same weight function $\frac{1}{1+\frac{b^2}{s^2}}$ up to a factor.

The parameters of the GTF are s and β ($\beta < 0$). They play exactly the same role as s and α in the SEF. For $-\frac{1}{2} \leq \beta < 0$, as previously the pdf is defined only for a bounded support.

The most surprising thing about this family is that, even if the tails of the pdf, which are controlled by β, can be very different, the weights are the same up to a factor β: $\frac{1}{1+t}$. The robust fitting is thus achieved by running exactly the same algorithm, whatever the value of β. However, β plays an important role in covariance matrix estimation as explained in Chapter 3.

2.3 Robustness Study

Following [30], in the fixed design case, the robustness of an M-estimator is characterized by its breakdown point, which is defined as the maximum percentage of outliers the estimator is able to cope with:

$$\varepsilon^*(\hat{A},Y,X) = \frac{1}{n}\min\{m \ : \sup_{\tilde{Y}\in B(Y,m)} \|\hat{A}(\tilde{Y},X)\| = \infty\} \tag{2.6}$$

where $\tilde{Y}$ is a corrupted data set obtained by arbitrary changing at most m samples (among the n samples of the data vector), B is the set of all $\tilde{Y}$: $B(Y,m) = \{\tilde{Y} \ : card\{k \ : \tilde{y}_k \neq y_k\} \leq m\}$ and $\hat{A}(\tilde{Y},X)$ is an estimate of A from $\tilde{Y}$. It is important to notice that the previous definition is different from the one proposed in [17] which is not suited to the fixed design setting.

Mizera and Müller [30] also emphasize the notion of *regularly varying* functions, and described the link between this kind of regularity and robustness property. By definition, f varies regularly if there exists a r such that:

$$\lim_{t\to\infty} \frac{f(tb)}{f(t)} = b^r \tag{2.7}$$

When the exponent r equals zero, the function is said to vary slowly, *i.e.* the function is heavily tailed.

We now assume that the ρ function of the M-estimator follows the four following conditions:

1. ρ is even, non decreasing on $\mathbb{R}^+$ and non negative,
2. ρ is unbounded,
3. ρ varies regularly with an exponent $r \geq 0$,
4. ρ is *sub-additive*: $\exists L > 0, \forall t,s \geq 0, \rho(s+t) \leq \rho(s)+\rho(t)+L$.

The main result proved in [30] is that the percentage ε^* is bounded by a function of r:

Theorem. *Under the four previous conditions on* ρ*, and if* $r \in [0,1]$*, then* $\forall$ *Y and X,*

$$\frac{M(X,r)}{n} \leq \varepsilon^*(\hat{A},Y,X) \leq \frac{M(X,r)+1}{n} \tag{2.8}$$

where, with the convention $0^0 = 0$*,* $M(X,r)$ *is defined as:*

$$M(X,r) = \min_{A\neq 0}\{card(K) : \sum_{k\in K} |X_k^t A|^r \geq \sum_{k\notin K} |X_k^t A|^r\} \tag{2.9}$$

where K runs over the subsets of $\{1,2,\cdots,n\}$.

When the exponent r is zero, the exact value of the percentage ε^* is known. The following theorem states its value.

Theorem. *Under the four previous conditions on* ρ *and if* $r = 0$*, then* $\forall\, Y$ *and* X*,*

$$\varepsilon^*(\hat{A}, Y, X) = \frac{M(X,0)}{n} = \frac{1}{n}\lfloor \frac{n - \mathcal{N}(X) + 1}{2} \rfloor \tag{2.10}$$

where $\lfloor x \rfloor$ *represents the integer part of* x*, and* $\mathcal{N}(X) = \max_{A \neq 0}\{card\{X_k : X_k^t A = 0\}, k = 1, \cdots, n\}$.

This value is also the maximum achievable value, which is approximately 50%. As a consequence, M-estimators with zero r exponent achieve the highest breakdown point.

Finally in [30], it is shown that the bounds on the percentage ε^* are related to r as a decreasing function. This is stated by the following lemma:

Lemma. *If* $q \geq r \geq 0$ *then*

$$M(X,q) \leq M(X,r) \leq M(X,0) = \lfloor \frac{n - \mathcal{N}(\mathcal{X}) + 1}{2} \rfloor.$$

These three results clearly establish the relationship between the exponent r of the function ρ and the robustness of the associated M-estimator. As a consequence of these important theoretical results, the robustness of a large class of M-estimators can be compared just by looking at the exponent r of the cologarithm of their associated noise pdf. To illustrate this, we now apply the above results to the SEF and GTF pdfs described in Section 2.2.

2.3.1 The SEF case

Let us show that the robustness of SEF based M-estimators is decreasing with $\alpha \in]0, 0.5]$. To this end, we first check the four above conditions on the ρ_α function. Function ρ_α is clearly even, non decreasing on $\mathbb{R}^+$ and non negative. The first condition is thus satisfied. The second one is fulfilled only when $\alpha > 0$, due to the fact that ρ_α is bounded for $\alpha \leq 0$. Looking at the ratio:

$$\frac{\rho_\alpha(tb)}{\rho_\alpha(t)} = \frac{(1 + t^2 b^2)^\alpha - 1}{(1 + t^2)^\alpha - 1} = \frac{(\frac{1}{t^2} + b^2)^\alpha - \frac{1}{t^{2\alpha}}}{(\frac{1}{t^2} + 1)^\alpha - \frac{1}{t^{2\alpha}}}$$

we see that when $\alpha > 0$, $\lim_{t \to \infty} \frac{\rho_\alpha(tb)}{\rho_\alpha(t)} = b^{2\alpha}$. As a consequence, the ρ_α function varies regularly and the third condition is also satisfied. For the fourth condition, we can use Huber's Lemma 4.2 [21], to prove the sub-additivity when $\alpha \in]0, 0.5[$. For $\alpha = 0.5$, it can also be proved that ρ_α is sub-additive. All conditions on ρ being fulfilled, the first Theorem applies, and using the Lemma, we prove that the robustness of SEF M-estimators is increasing towards the maximum of approximately 50%, with respect to a decreasing α parameter within $]0, 0.5]$.

2.3.2 The GTF case

Let us prove that the robustness of GTF M-estimators is maximum, whatever the value of β. We shall first check the 4 conditions on the associated ρ function. The first two assumptions are easy to check for ρ_β. Looking at the ratio:

$$\frac{\rho_\beta(tb)}{\rho_\beta(t)} = \frac{\ln(t^2) + \ln(\frac{1}{t^2} + b^2))}{\ln(t^2) + \ln(\frac{1}{t^2} + 1))}$$

we deduce: $\lim_{t\to\infty}\frac{\rho_\beta(tb)}{\rho_\beta(t)} = 1 = b^0$. As a consequence, the ρ_β function varies slowly, and the third condition is fulfilled. The fourth condition is proved by using Huber's Lemma 4.2 [21]. All conditions on ρ being satisfied, the second Theorem applies, showing that GTF M-estimators achieve the highest breakdown point of approximately 50%.

2.4 Noise Model Parameters Estimation

In the previous section, two noise models were proposed, which can achieve the 50% breakdown point. To use them in practical applications, it remains to estimate the scale s and the power α or β of the models from residuals obtained in images.

When the pdf support is unbounded and other noise parameters are assumed to be known, by deriving the negative log-likelihood w.r.t. s, we get the following implicit equation in $\hat{s}$:

$$\hat{s}^2 = \frac{1}{n}\sum_{i=1}^{i=n}\phi'\left(\frac{(X_i^t A - y_i)^2}{\hat{s}^2}\right)(X_i^t A - y_i)^2 \tag{2.11}$$

This equation means that the MLE estimate $\hat{s}$ of s is a fixed point. Other scale estimators such as MAD [17], have been proposed. In the case of a bounded support, a more complex equation, involving the integral of the pdf on its support, can also be derived.

Interestingly, when the stretched exponential family (2.2) is used, (2.11) is simplified and gives the following closed-form estimator:

$$\hat{s} = \left(\frac{2\alpha}{n}\sum_{i=1}^{i=n} b_i^{2\alpha}\right)^{\frac{1}{2\alpha}} \tag{2.12}$$

The MLE estimator of the scale must be used with care. Surprisingly, for low noise levels, we observed that the scale is clearly under-estimated. This is a general problem, due to the finite precision of feature detectors (in our case, it provides discrete image positions). In Figure 2.3, we simulate this effect by generating a Cauchy noise ($\beta = 1$) with increasing scales, and rounding it. Then, the MLE scale $\hat{s}$ estimate is computed from this data, assuming a Cauchy noise with unknown scale in (2.11). Figure 2.3 confirms our observation: when the true scale s is lower than one pixel, it is clearly under-estimated.

This suggests that during its estimation, the scale must not been allowed to take small values (*i.e.* lower than one pixel). This also implies that it is better not to estimate the scale between each step of the robust fitting, in contrast to what was suggested in [20]. Indeed, an under-estimated scale can freeze the fit in its current state, while it has not yet converged, as explained in Section 2.6.

The MLE approach, when applied to the estimation of α or β, assuming a known s, does not seem to yield close-form estimators. Nevertheless, this estimation may always be performed by numerically minimizing the likelihood w.r.t. α, using for example a Gradient Descent algorithm.

The MLE approach can also be applied to estimate α and s at the same time by a non-linear minimization as shown in Figure 2.4 where residuals are collected from a set of 150 images. In other experiments, we also typically obtained small scale estimates, due to the feature extractor, and values of α near zero. Since s is relatively small compared to the size of the image, the quantized data provided by the lane-marking extractor may thus produce an under-estimated s.

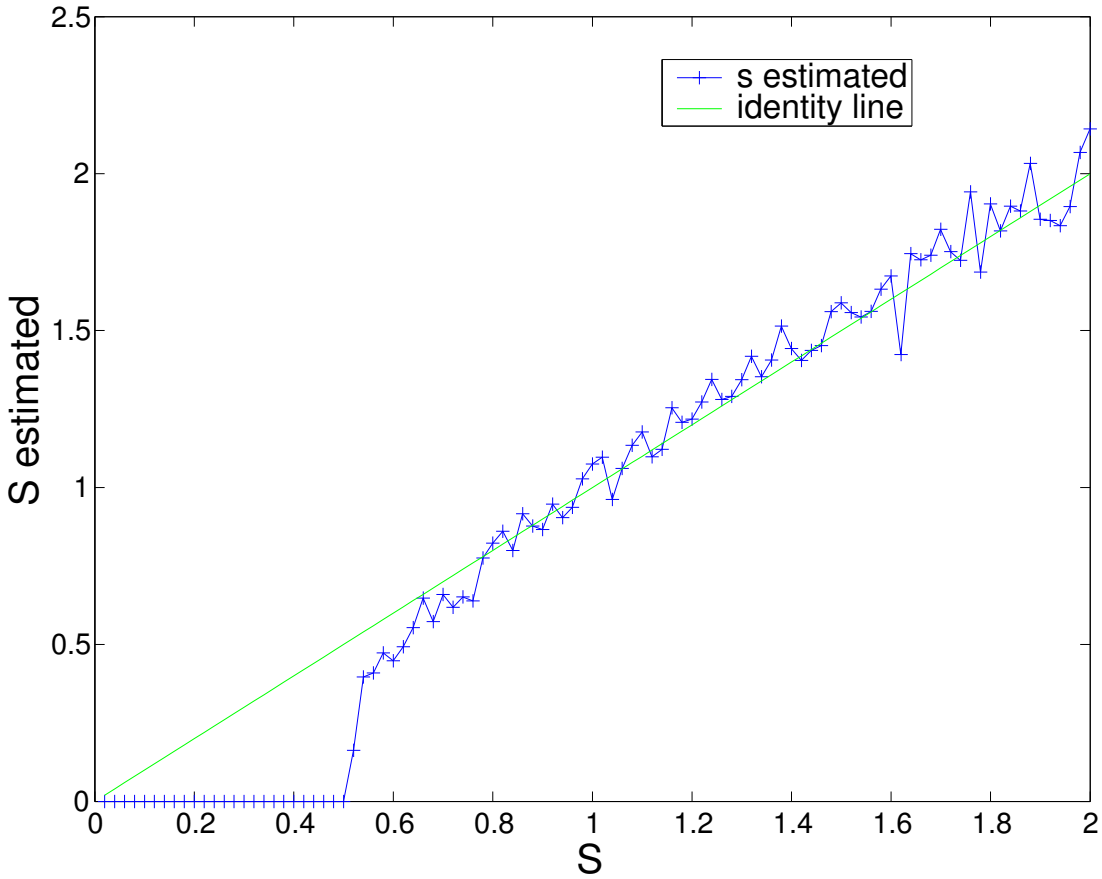


Figure 2.3: *The estimated scale $\hat{s}$ versus the true scale s for Cauchy noise. We can notice that s is under-estimated by the MLE estimator $\hat{s}$ when $s < 1$, due to data rounding*

As for the MLE estimator of s, the MLE estimator of (α, s) must be used with caution. Indeed, there are singular sets of residuals were α and s can not be estimated properly.

Finally, to estimate $(\hat{s}, \hat{\alpha})$, we can envision three different approaches to collect the residuals:

1. Collect residuals from a large set of images that are statistically representative of the application. Estimate the noise parameters and use them as fixed values in the curve fitting algorithm.

2. Collect residuals from each image after fitting (using averaged noise parameters). Refine the curve fit with the obtained noise model parameters.

3. When processing an image sequence, collect residuals from the m previous images and use them to estimate the noise parameters. Then perform curve fitting on the current image using the previously estimated parameters, assuming a temporal continuity.

The first approach seems to be convenient for pure recognition or detection problems in a single image. But when tracking is involved, noise parameters are not constant in time. Thus,

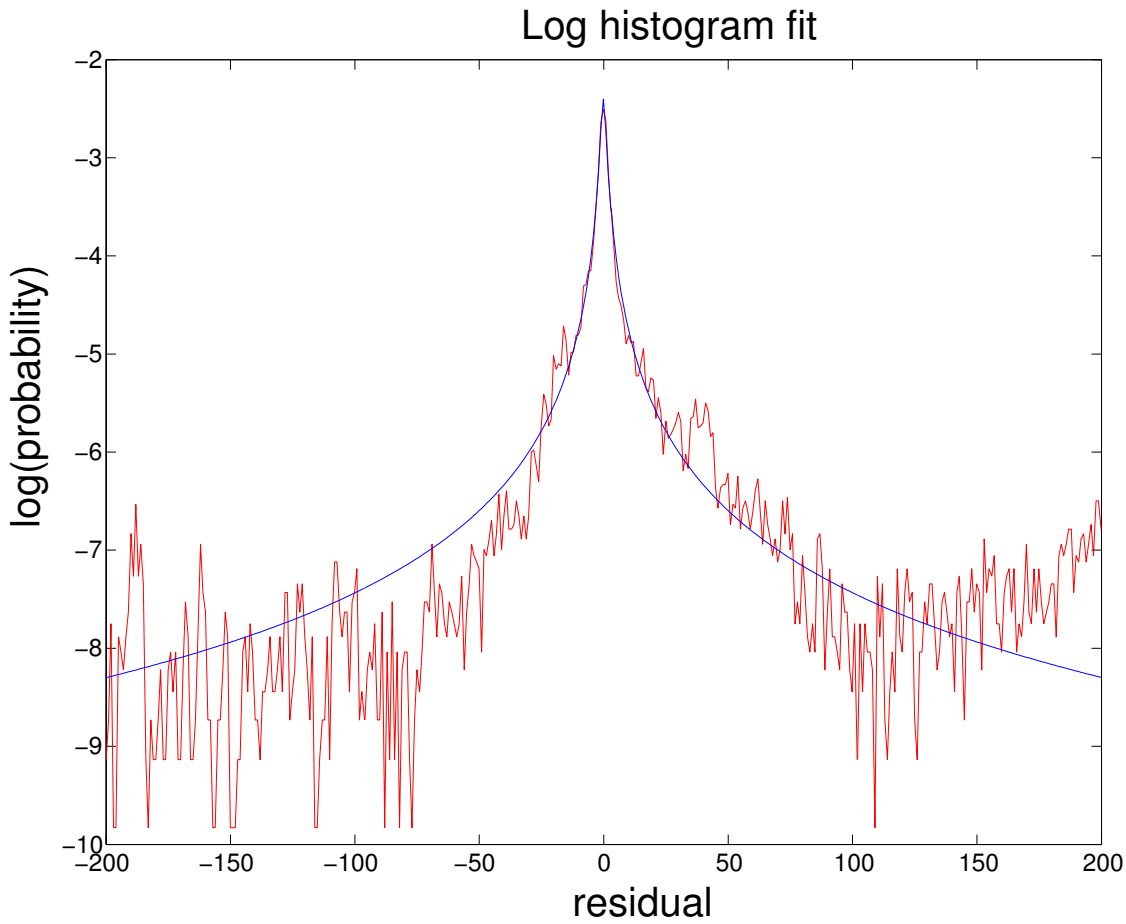


Figure 2.4: *Log histogram of the residual errors in feature position collected from* 150 *images after lane-marking fits. With the MLE approach, this distribution is well approximated by the best fit in the SEF (2.3) with parameters* $\alpha = 0.05$ *and* $s = 1.9$

the second approach seems more attractive. Nevertheless, it may lead to bad estimates of the noise parameters in case of an image without lane-marking. Therefore from our experiments, we conclude that the third approach should be adopted.

2.5 Sensitivity of Fitting to Noise Model Parameters

By their very nature, robust fitting algorithms are not very sensitive to outliers. However, another question of importance is their robustness to an inappropriate choice of the noise model parameters. In order to evaluate the effect of this kind of modeling error on the fitting, we ran the following experiment:

1. We collect a set of 150 images showing the same lane-marking in the same position but with different perturbations (some of which are shown in Figure 2.5).

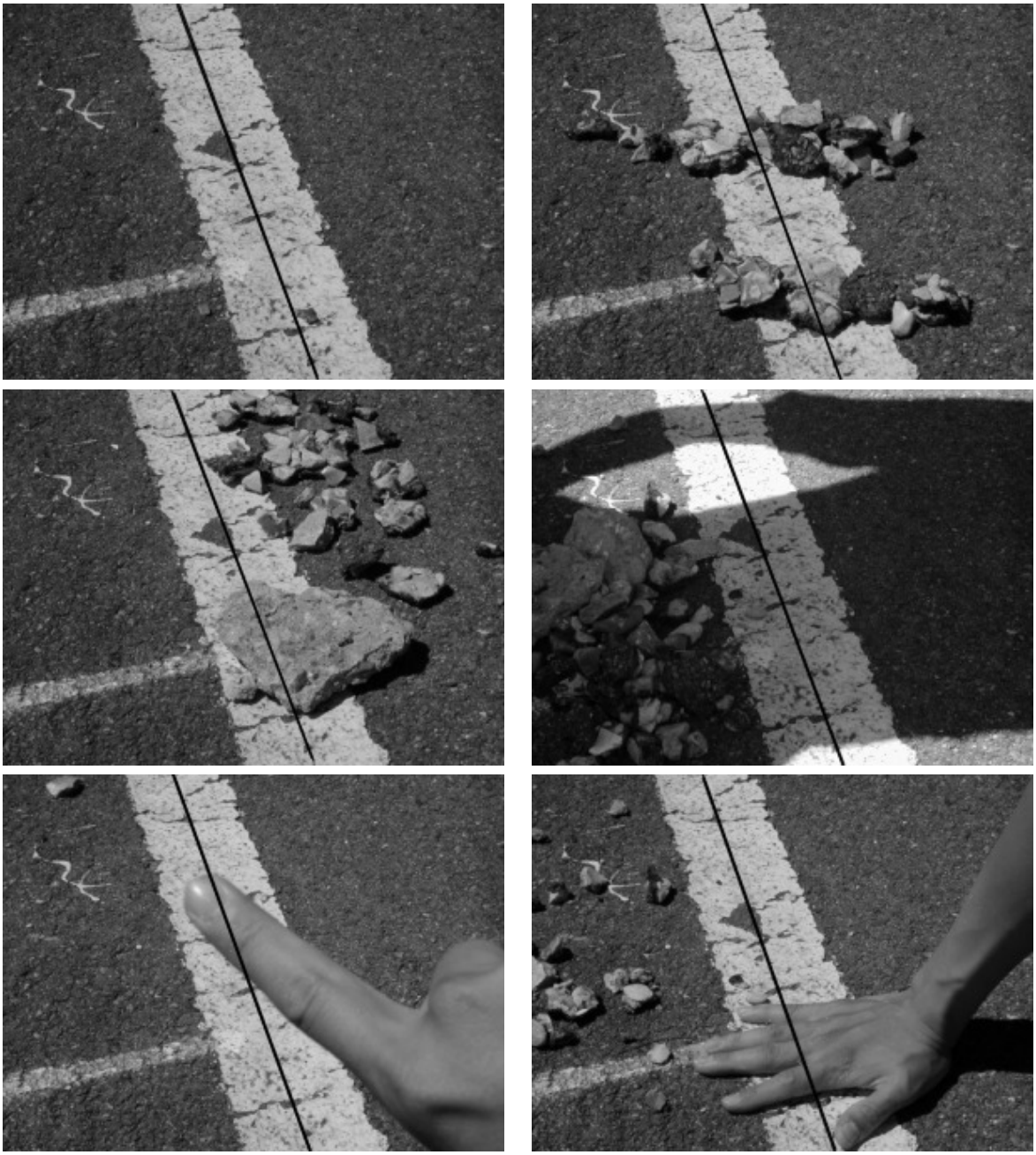


Figure 2.5: *Six of the* 150 *images used in the experiments. The black straight line is the reference fit*

2. The reference lane-marking was accurately measured by hand. In this experiment fits are straight lines.

3. For each image of the set, fits were estimated for pairs of (α, s) on a grid assuming a noise model in the smooth exponential family.

4. The relative error between the fits and the reference fit was averaged over the image set for each (α, s). Figure 2.6 shows the obtained results.

On the top of Figure 2.6, with a random initialization, we observed that the valley along a α cross-section is very wide compared to an s cross-section. Moreover, when the initialization is set in the vicinity of the reference as on the bottom of Figure 2.6, the error function between the fit and the reference for different α and s is very flat for $\alpha < 0$. This is important to notice, since it means that when α is chosen in the range of correct values, s can vary over a large range without really modifying the resulting fits. This can also be understood by the fact that the SEF (2.3) behaves approximately like the stretched exponential family (2.2). And as noticed in [3], the interesting fact with the stretched exponential family is that the obtained fits are invariant with respect to the value of s. Indeed, the weights computed in step 2 of the robust fitting algorithm are $\lambda_i = 2\alpha \frac{b_i^{2(\alpha-1)}}{s^{2(\alpha-1)}}$. Thus, step 3 is not modified whatever the value of s, since $s^{2(\alpha-1)}$ can be simplified on the left and right hand sides after substitution of the weights. Least Squares Fitting shares the same invariance property as explained in Section 1.6.1. A similar argument can be used to show that the robust fitting algorithm is invariant to β when the noise model is chosen in the generalized T-Student family.

To summarize, the robust fitting algorithm with a noise model chosen from the SEF is, in our context, relatively robust to the scale parameter s but less robust to errors on the power parameter α. With the GTF noise model, the robust fitting algorithm is invariant to the choice of β.

2.6 Improving Convergence: the GNC Strategy

The relative robustness of the curve fitting observed in the previous section can be exploited to speed up the convergence toward a local minimum and to help select a local minimum close to the global one.

Figure 2.7 shows that the weight λ, used in the IRLS algorithm, becomes more sharply peaked when α or s decrease. As a consequence, the lower α or s, the lower the effect of outliers on the result and thus, the more robust the fitting.

Thus, when α decreases, the robust error function $e_R(A)$ becomes less and less smooth. If $\alpha = 1$, the cost function is a paraboloid and thus there exists a unique global minimum. By decreasing α to values lower than $\frac{1}{2}$, local minima appear (when curve is with more than one parameter, $d+1 > 1$). This is illustrated in Figure 2.8 where the robust error function $e_R(A)$ is shown for three decreasing values of α.

As explained in [2], the localization property of the robust fitting w.r.t. decreasing α can be used to converge toward a local minimum close to the global one. This approach, called Graduated Non Convexity (GNC), consists in, first, enforcing local convexity on the data using a large value of α. Then, a sequence of fits with decreasing α, is performed in continuation,

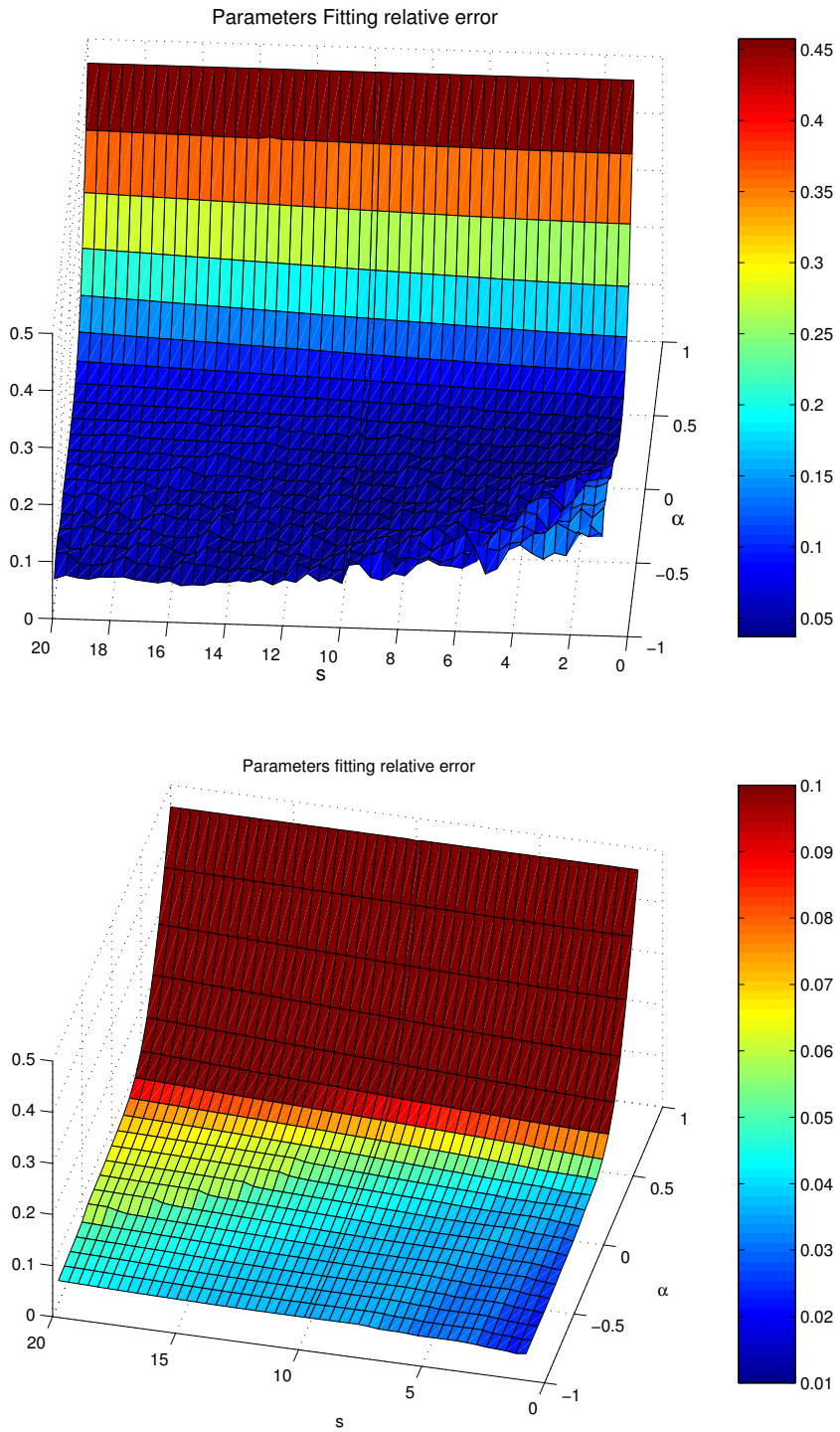


Figure 2.6: *Errors between the fit and reference for different* α *and s. Top, the initial curve parameters* A_0 *are set to random values; bottom,* A_0 *is set in the vicinity of the reference. Notice, how broad the valley in s is when the value of* α *is properly chosen*

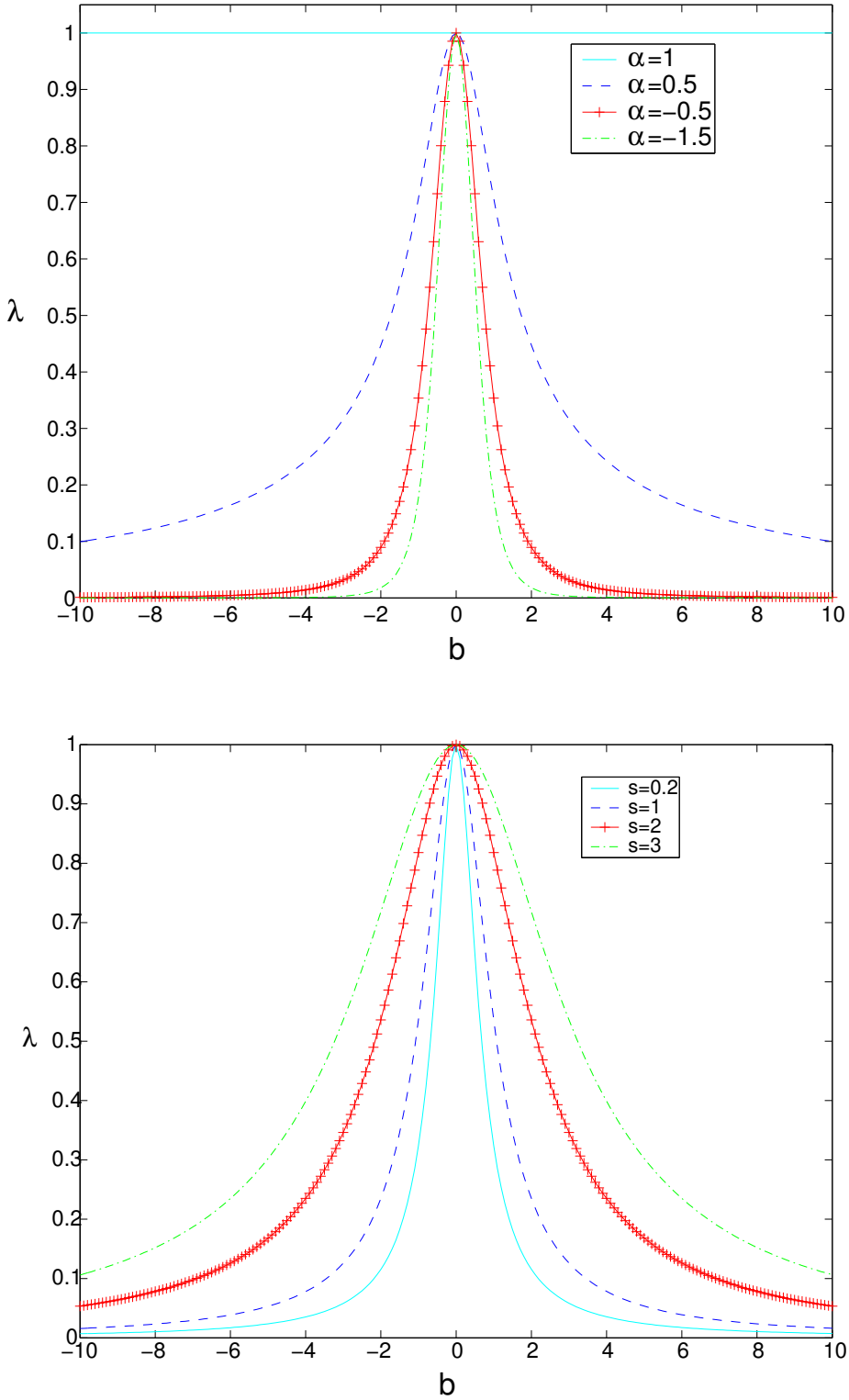


Figure 2.7: *Variations of weight* λ *with respect to the residual error* b*, top for* $s = 2.4$ *and different values of* α*, and bottom for* $\alpha = 0.1$ *and different values of* s

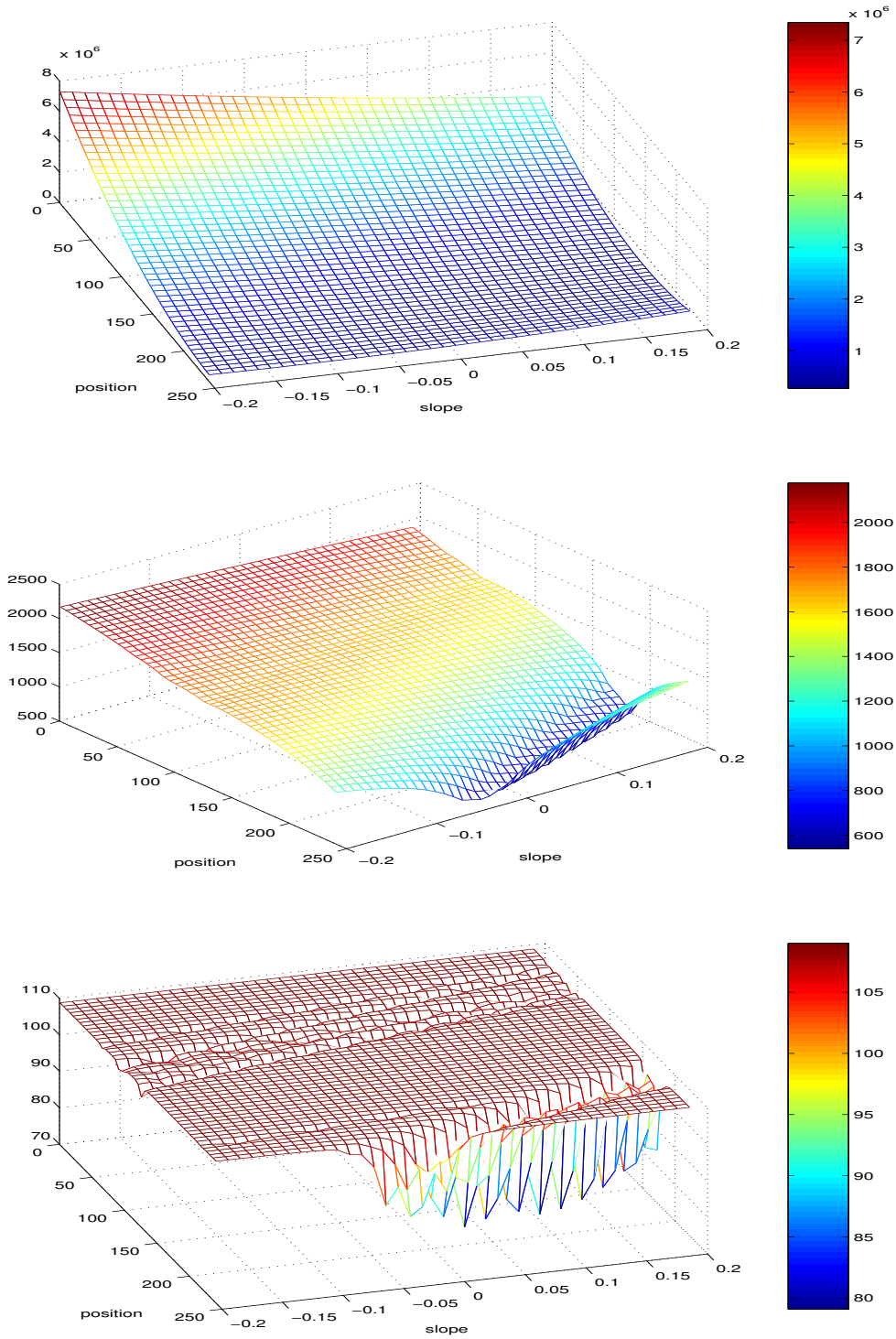


Figure 2.8: *The robust error function $e_R(A)$ for the two parameters of a straight line for different values of α: Gaussian noise ($\alpha = 1$), near T-Student noise ($\alpha = 0.1$), Geman&McClure noise ($\alpha = -1$), from top to down*

i.e. each time using the current output fit as an initial value for the next fitting step. But one must be careful not to decrease α or s too fast. Indeed, with too small an α or s, the number of local minima increases and the curve fitting algorithm has a higher chance of being trapped into a local minimum located far from the global one. In practice, this implies that the estimated curve parameters are *frozen* at their current values during the alternate minimization [9]. As a conclusion, the lower bound of the decreasing α must not be too small compared to the $\hat{\alpha}$ estimated from the residuals as explained in Section 2.4.

An increasing scale s has the same smoothing effect on the robust error as an increasing α. Thus, as proposed by [32], large scales can be used similarly to high α to converge to a local minimum that is close to the global one and to speed up the convergence. On the other hand, when too small a value of s is chosen, the curve parameters can also be frozen to their current values during the fit. Therefore, to maintain good convergence properties and to have an algorithm that is as fast as possible, the MLE estimate $(\hat{s}, \hat{\alpha})$ helps us to select the correct values for the lower bounds of a sequence of decreasing s and α.

2.7 Conclusion

In this chapter, the important question which we have addressed is the *choice of the noise distribution model parameters*. We have introduced two-parameter families of noise models that allow a continuous transition between the Gaussian model and non Gaussian models. We have showed how this last property can be used to improve fitting results and robustness: even with a slightly incorrect noise model of these families, the estimated fits remain relatively reliable.

Moreover, we applied Mizera and Müller's fundamental work on M-estimators breakdown point calculation in the field of image analysis. In the *fixed design* setting, they shown that certain M-estimators can achieve maximum robustness. Using their results, we discussed the robustness of M-estimators based on two non-Gaussian pdfs families, that we introduced under the names of SEF and GTF. In particular, we showed that the GTF noise model leads to M-estimators that achieve the maximum breakdown point of approximately 50%, and that the robustness associated with SEF increases towards the maximum as α decreases towards 0.

To finish, we illustrated how useful these results are for lane-marking detection: SEF and GTF approximate models seems to correctly fit observed noise pdfs. The sensitivity to noise parameters appears to be good, especially when using the GNC approach. We therefore believe that the SEF and GTF families can also be used with advantages in many other image analysis algorithms.

Chapter 3

Evaluating the Fits and Tracking

3.1 Introduction

In practical applications, an estimate must be accompanied by some measure of *confidence* in order to be correctly used by higher level systems, such as Kalman filters in tracking applications. In the statistical estimation framework, this confidence measure is naturally given by (the inverse of) the covariance matrix. That is the main reason why we formulated detection as a fitting problem in Chapter 1. With Gaussian noise, computing the covariance matrix is straightforward, see Section 3.2.

In the robust framework, however, a correct covariance matrix estimate is more difficult to obtain. There are two main reasons for this. First, there is no known closed-form solution, but only approximations. In this chapter, we focus on approximations that can be quickly computed from the data in hand during the fitting. Second, it turns out that an accurate choice of the noise model is much more critical for the covariance matrix estimation than for the fitting. Despite these difficulties, we are able to propose fast-to-compute and efficient approximates of the covariance matrix as explained in Section 3.3.

In the last two sections, for an intuitive understanding, we make a gradual presentation of the robust Kalman framework which is useful for implementing tracking along an image sequence. In Section 3.4, we introduce recursive and robust least squares (recursive least squares is a simple case of Kalman filter, using a constant state model). Finally in Section 3.5, the robust Kalman filter is described. This Chapter was partially published in [40, 23].

3.2 Exact Covariance Matrix for Least Squares Fitting

Using notations of Section 1.6.1, the covariance matrix C_{LS} of Least Squares Fit A_{LS} is $E[(A_{LS} - E[A_{LS}])(A_{LS} - E[A_{LS}])^t] = S^{-1}XE[(Y - E[Y])(Y - E[Y])^t]X^tS^{-t}$. We have $E[(Y - E[Y])(Y - E[Y])^t] = \sigma^2 I_{n\times n}$, since the noise b is iid with variance σ^2, where $I_{n\times n}$ denotes the $n \times n$ identity matrix. Finally, the inverse covariance matrix of A_{LS} is deduced as:

$$C_{LS}^{-1} = \frac{1}{\sigma^2} S = Q_{LS} \tag{3.1}$$

Q_{LS} is also known as Fisher's information matrix for the set of n data points. Q_{LS} is defined as the expectation of the second derivative of e_{LS} w.r.t A.

Finally, it is important to notice an interesting property of Least Squares fitting: the fit does not rely on a correct estimate of σ. σ is requested only for computing the covariance matrix. Therefore, σ is generally estimated just before covariance matrix computation, from the residuals obtained after parameter fitting. The MLE estimator of σ is obtained by canceling the first derivative w.r.t σ of $-\ln(p)$, which yields:

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{i=n}(X_i^t A_{LS} - y_i)^2 \tag{3.2}$$

3.3 Approximate Confidence Matrices for Robust Fitting

A complete review of the many different approaches for approximating covariance matrices is beyond our scope here. Thus, we present a selection of five matrices that can be efficiently computed. Then, we propose a new approximation whose performances in terms of quality will be evaluated.

3.3.1 Available Approximates

The simplest approximate covariance matrix C_{Cipra}, named after Cipra's covariance matrix from [40], is:

$$C_{Cipra} = s^2 \left(\sum_{i=1}^{i=n} \lambda_i X_i X_i^t\right)^{-1},$$

where λ_i are the values taken by the weights after robust fitting converged. Another fast-to-compute approximation is proposed in [40]:

$$C_{Simple} = s^2 \left(\sum_{i=1}^{i=n} \lambda_i^2 X_i X_i^t\right)^{-1}.$$

Notice that in these two definitions, s^2 appears as a factor. As a consequence, incorrect values of the scale parameter lead to poor-quality estimates.

In chapter 7 of [20], Huber derives an asymptotic covariance matrix and proposes three other approximate covariance matrices:

$$C_{Huber1} = K^2 \frac{\frac{1}{n-d-1}\sum_{i=1}^{i=n}(\rho'(b_i))^2}{(\frac{1}{n}\sum_{i=1}^{i=n}\rho''(b_i))^2}\left(\sum_{i=1}^{i=n} X_i X_i^t\right)^{-1},$$

$$C_{Huber2} = K \frac{\frac{1}{n-d-1}\sum_{i=1}^{i=n}(\rho'(b_i))^2}{\frac{1}{n}\sum_{i=1}^{i=n}\rho''(b_i)}\left(\sum_{i=1}^{i=n} \rho''(b_i) X_i X_i^t\right)^{-1},$$

$$C_{Huber3} = K^{-1}\frac{1}{n-d-1}\sum_{i=1}^{i=n}(\rho'(b_i))^2 W^{-1}\left(\sum_{i=1}^{i=n} X_i X_i^t\right)W^{-1},$$

with Huber's notation $\rho(t) = \phi(t^2)$ and $W = \sum_{i=1}^{i=n} \rho''(b_i) X_i X_i^t$. n is the number of data points and $d+1$ is the dimension of vectors A and X_i. The correction factor, K, is given by:

$$K = 1 + (d+1) \frac{\sum_{i=1}^{i=n} (\rho''(b_i) - \frac{1}{n} \sum_{i=1}^{i=n} \rho''(b_i))^2}{(\sum_{i=1}^{i=n} \rho''(b_i))^2}.$$

For easier notations, we set $O_1 = \sum_{i=1}^{i=n} \lambda_i X_i X_i^t$ and $O_2 = \sum_{i=1}^{i=n} \lambda_i^2 X_i X_i^t$. The inverse of a covariance matrix will be called a *confidence matrix* in the following.

We propose the following new approximate covariance matrix which is justified in the Appendix 6.2:

$$C_{ITC} = \frac{\sum_{i=1}^{i=n} \lambda_i b_i^2}{\sum_{i=1}^{i=n} \lambda_i - Trace(O_2 O_1^{-1})} O_1^{-1} O_2 O_1^{-1}. \tag{3.3}$$

This approximation is not an asymptotic covariance matrix, unlike Huber's approximations. The advantage is thus that C_{ITC} does not require a high number n of data points to be applied. It relies on assuming non-random weights λ_i. Indeed, λ_i is bounded between 0 and 1, and thus may be assumed to have a smaller variance than b_i. If the weights were considered as random variables, deriving the covariance matrix would become too involved.

3.3.2 Confidence Matrix Comparison

Our first test consists in comparing the approximate confidence matrices one to another, for straight-line fitting, on the 40 first images of the 150 real images partly shown in Figure 2.5. Figure 3.1 displays the values of the components of one confidence matrix approximate versus another one. The three different components of the 2×2 symmetric confidence matrix are shown with different colors. Figure 3.1 illustrates the fact that the approximate confidence matrices are approximately linearly correlated. This means that the different approximates are in fact not so different, contrary to what was said in [23].

Our second test on real images is performed for each of the six approximate confidence matrices but, this time, with respect to a reference confidence matrix. The reference covariance matrix is obtained by estimating the covariance matrix of the fits on 150 real images. The experimental comparison setup is the same as in Section 2.5, with two additional steps:

5. The six approximate covariance matrices presented are computed for each image.

6. The reference matrix is computed as the covariance matrix of the fits. The relative errors between the reference and the six approximate covariance matrices are computed on the image set for each (α, s)

The error measure that we use is an average over the images and the components of the ratio of the absolute difference between the estimate and the reference over their mean:

$$\frac{1}{(d+1)^2} \sum_{i=0}^{d} \sum_{j=0}^{d} \frac{| C_{ij} - C_{ref,ij} |}{\frac{C_{ij} + C_{ref,ij}}{2}}. \tag{3.4}$$

With such a definition, the relative error is always between 0% and 200%. 0% corresponds to the best case, where the matrices are identical, while 200% corresponds to the worst case,

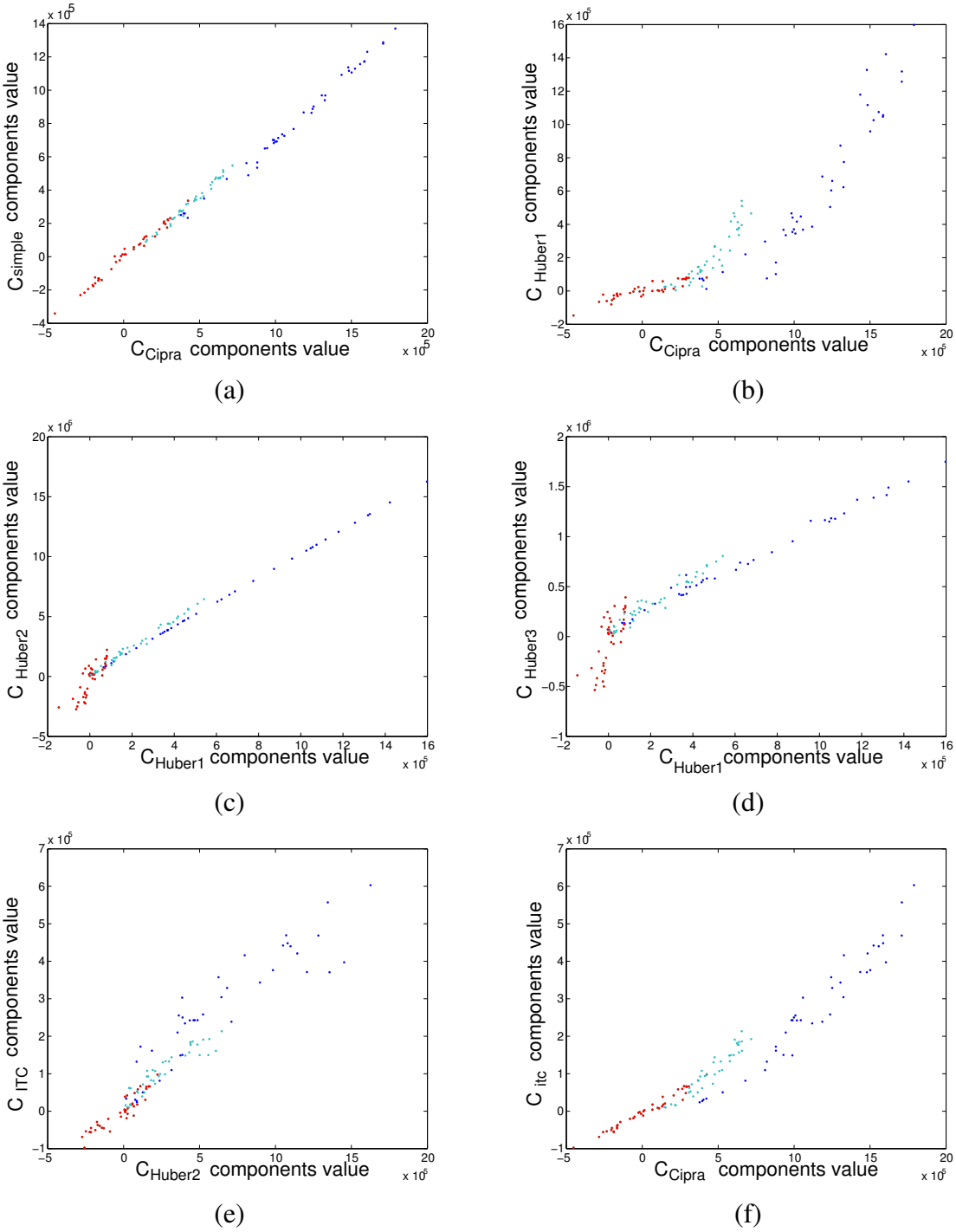
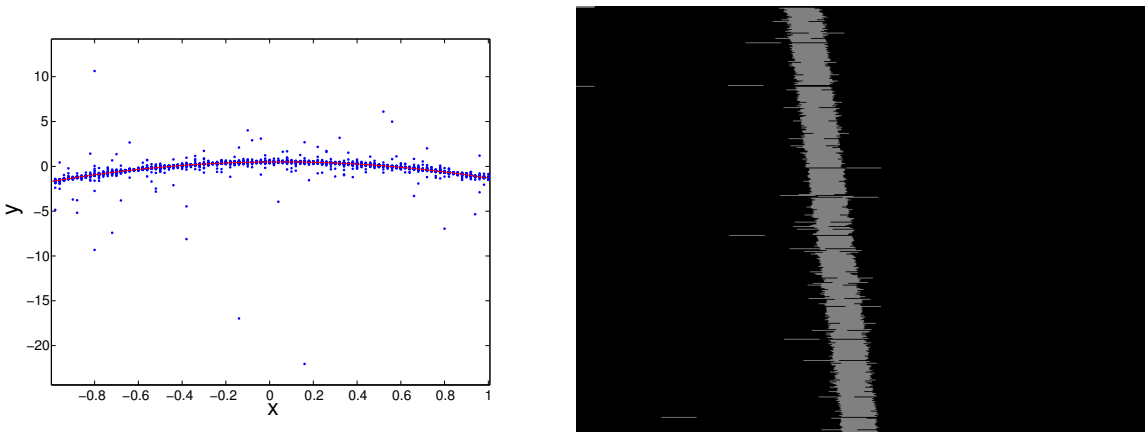


Figure 3.1: *Plots of the three components of the* 2×2 *symmetric confidence matrices (each one being shown with a different color). Each point of a plot corresponds to values computed from one of the 40 sample images: the abscissa is the value of the component in one confidence matrix while the ordinate is the value of the same component in another confidence matrix. The following comparisons are made: (a) is for* C_{simple} *versus* C_{cipra}*, (b) for* C_{Huber1} *versus* C_{cipra}*, (c) for* C_{Huber2} *versus* C_{Huber1}*, (d) for* C_{Huber3} *versus* C_{Huber1}*, (e) for* C_{ITC} *versus* C_{Huber2}*, (f) for* C_{ITC} *versus* C_{Cipra}

where matrix C is very far from the reference one C_{ref}. To fix the ideas: a 50% relative error means that C and C_{ref} are related by a multiplicative factor of about 1.67. A 100% relative error corresponds to a factor of 3, 150% corresponds to a factor of 7 and 180% to slightly more than 20.

The relative errors w.r.t. the reference matrix for the different approximate covariance matrices were compared, for $(\alpha, s) = (0.05, 1.9)$, i.e. the value obtained in Fig. 2.4, but results turns to be poor. Similar poor results are obtained for other (α, s) pairs in the "valley" of Figure 2.6. Indeed the relative errors are around 180%.

3.3.3 Comparison on Synthetic Data



(a) (b)

Figure 3.2: *(a) 100 random samples of noisy second order polynomial curve.*
(b) sample of a synthetic marking image with Cauchy noise

To understand why such poor results were obtained on real images, we ran extra experiments on synthetic data, to see if covariance approximates are consistant when data agree with the assumed models. 20000 data sets perturbed with Cauchy noise were generated and fitted by a second degree polynomial assuming the true noise model parameters. Fig. 3.2(a) shows examples of noisy data sets.

Tab. 3.1 displays the statistics, over the 20000 fits, of the average relative errors (3.4) for the 6 approximate confidence matrices. From this table, it appears that all matrices give similar relative errors. This last observation is consistent with what was observed on real data. It also appears that even if the data fully agree with the assumed models, the obtained relative errors are around 80%. The corresponding multiplicative factor is one order of magnitude less than in the real data case. In practice, such kind of relative errors provide usable enough estimates.

To find out if the difference of results between synthetic and real data are due to the *extraction* algorithm (Sec. 1.4), we generate synthetic images of a straight lane-marking perturbed by Cauchy noise on each line. As shown in Fig. 3.2(b), the obtained images are quite realistic. We rerun the previous experiment, using 200 of these synthetic images and including the extraction algorithm in the process. The results are similar to what was obtained with synthetic noisy

Table 3.1: Mean and standard deviation of the relative errors (in percentage) w.r.t. the reference, see (3.4), for the six covariance matrix estimates, with the true noise model. These statistics are computed from 20000 fits.

Type	relative error
Cipra	$65\% \pm 27\%$
Simple	$72\% \pm 23\%$
Huber1	$86\% \pm 17\%$
Huber2	$80\% \pm 28\%$
Huber3	$87\% \pm 28\%$
ITC	$79\% \pm 33\%$

curves, which mean that perturbations introduced by the marking extractor cannot explain the poor results obtained on real images.

3.3.4 Correlated Data

After investigation on real data, we observed that the reference and estimated covariance matrices differ mainly by a multiplicative factor. This leads us to question of the noise model. Indeed, our underlying assumption is that the noise is independent. However, due to road texture, the noise is correlated from one image line to the other. For the Gaussian noise case, the effect of a correlation can be approximated as an extra multiplicative factor on the confidence matrix. We propose to extend this approximation to the Robust case by introducing an *ad hoc* correction factor on the covariance matrix approximates. All previous approximates can thus be improved by this correction factor, which accounts for data noise correlations in the confidence matrix and which can be written as:

$$1 - \frac{\sum_{i=1}^{i=n-1} \sqrt{\lambda_i w_i \lambda_{i+1} w_{i+1}}}{\sum_{i=1}^{i=n} \lambda_i w_i}$$

using notations of Section 1.6.2.

To assess the appropriateness of this correction factor, we experiment using synthetic images where the noise is correlated along image lines, as in Fig. 3.3(a)(b). Without the correction factor, the relative error on the C_{ITC} confidence approximate is $154\% \pm 35\%$. This corresponds to a too large relative error. When the correction factor is used, we almost reach the same performances as with non-correlated synthetic data: the relative error drops to $92\% \pm 49\%$.

These results were confirmed on 200 real images, similar to those shown on Fig. 3.3(c)(d), featuring a real lane marking with randomly placed pebbles. We obtained that the relative error on the C_{ITC} confidence approximate, without correction factor, is a multiplicative factor within $[0.058, 1.611]$. When the correction factor is used, the relative error becomes closer to one since it takes on value within $[0.363, 1.171]$.

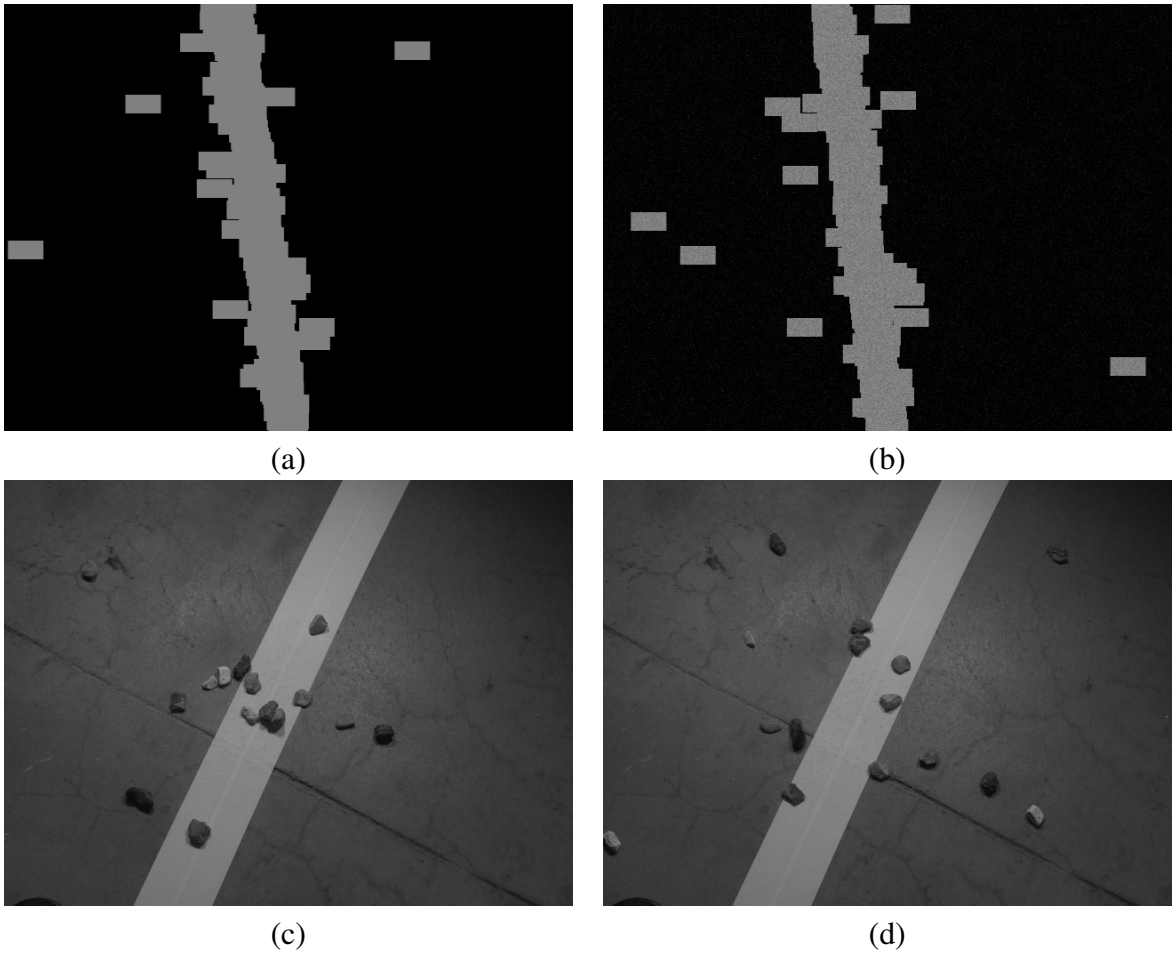


Figure 3.3: *(a) sample of a synthetic marking image with Cauchy noise correlated along lines ; (b) another similar sample where random noise is added on the intensity ;*
(c) and (d) images of a real lane marking with randomly placed pebbles.

3.4 Recursive Fitting

We now consider the problem of sequentially processing images. The steady-state situation consists in supposing that we observe, at every time step t, the same underlying curve. Suppose that images are indexed by t and that for each image, we have to fit its n_t data points $(x_{i,t}, y_{i,t})$, $i = 1, ..., n_t$. It is important to take advantage of the consistency along time to introduce some interpolation in case of missing parts of discontinuous lane markings. Such a consistency assumption can be implemented in the form of a prior bias from the past when fitting the data available on the current image. Sequential processing also saves memory space and computation time compared to global processing, allowing real-time implementation, which is crucial for driver assistance systems.

3.4.1 Recursive Least Square Fitting

In the least squares framework, recursive algorithms are based on an exhaustive summary of the data points, observed before t. Indeed, the error of the data points from time 1 to t is:

$$e_{rLS,t}(A) = \frac{1}{2\sigma^2} \sum_{k=1}^{k=t} \sum_{i=1}^{i=n_t} (F(x_{i,k})^t A - y_{i,k})^2$$

This sum can be rewritten as the sum of the error at time t alone and of the error from time 1 to $t-1$:

$$e_{rLS,t}(A) = \frac{1}{2}(A - A_{rLS,t-1})^t Q_{rLS,t-1}(A - A_{rLS,t-1}) + \frac{1}{2\sigma^2} \sum_{i=1}^{i=n_t} (X_{i,t}^t A - y_{i,t})^2 \tag{3.5}$$

Using (1.4), the summary of the past error consists in the previously fitted solution $A_{rLS,t-1}$ and its Fisher's matrix $Q_{rLS,t-1}$. By comparing $e_{rLS,t}$ with e_{LS}, the exhaustive summary by $A_{rLS,t-1}$ and $Q_{rLS,t-1}$ can be interpreted as a Gaussian prior on A at time t.

The error $e_{rLS,t}$ is quadratic and using (1.4) its second order matrix is $Q_{rLS,t}$. Taking second derivative of (3.5), we deduce:

$$Q_{rLS,t} = Q_{rLS,t-1} + \frac{1}{\sigma^2} S_t \tag{3.6}$$

where $S_t = \sum_{i=1}^{i=n_t} X_{i,t} X_{i,t}^t$. The recursive update of the fit is obtained by solving the following linear system obtained by canceling the first derivative of $e_{rLS,t}$ with respect to A:

$$Q_{rLS,t} A_{rLS,t} = Q_{rLS,t-1} A_{rLS,t-1} + \frac{1}{\sigma^2} T_t \tag{3.7}$$

with $T_t = \sum_{i=1}^{i=n_t} y_{i,t} X_{i,t}$. As a consequence, the recursive fitting algorithm consists of the following steps:

1. Initialize the recursive fitting by setting $Q_{rLS,0}$ and $A_{rLS,0}$ to zero, and set t=1.
2. For the data set associated with step t, compute the matrix $S_t = \sum_{i=1}^{i=n_t} X_{i,t} X_{i,t}^t$ and the vector $T_t = \sum_{i=1}^{i=n_t} y_{i,t} X_{i,t}$ only related to the current data set.
3. Update the Fisher's matrix $Q_{rLS,t}$ using (3.6).
4. Compute the current fit $A_{rLS,t}$ by solving the linear system (3.7).
5. If a new data set is available, increment t and go to 2.

The solution, obtained by this recursive algorithm at step t, is the same that the one obtained by standard least squares using all points of time steps from 1 to t. It is the so-called recursive (or sequential) least squares algorithm (subscript rLS). Note that no matrix inverse is explicitly needed. Only one linear system is solved at every time step t. This is interesting in real-time applications, since the complexity for solving the linear system is $O(d^2)$, when it is $O(d^3)$ for matrix inversion.

Note that (3.6) gives the recursive update of the Fisher's matrix $Q_{rLS,t}$ as a function of the previous Fisher's matrix $Q_{rLS,t-1}$ and of the current scatter matrix S_t. A better initialization of $Q_{rLS,0}$ than 0 consists in β times the identity matrix, where β has a positive value close to zero. This initialization insures that the solution of (3.7) is unique. Indeed, $Q_{rLS,t}$ is definite for any t, even if S_t in (3.6) is not. This is equivalent to the *Ridge Regression* regularization [42]. More generally, $Q_{rLS,0}$ is the inverse covariance matrix on the Gaussian prior on the curve parameters A, leading to a Maximum A Posteriori (MAP) estimate.

3.4.2 Recursive Robust Fitting

Is it possible to generalize this recursive scheme in the robust case? In general, an exact answer is negative: it is not possible to summarize the past in a few matrices, except when the noise distribution belongs to a particular kind of function, called *exponential family* in the probabilistic literature (see sufficient statistics in [34]). Unfortunately, the stretched exponential family (2.2), SEF and GTF do not belong to this family. As a consequence, to obtain a solution without approximation with SEF and GTF, the computation of the weights λ_i would require storing all past observed points in memory up to the current time step. In practice, this is a problem, since it means that the number of computations will increase with time t. Clearly, a second level of approximation is needed - remember that the first one consists in the approximate computation of the inverse covariance matrix as described in Section 3.3. It is usual to consider $A_{rR,t-1}$ as Gaussian with a covariance matrix $C_{rR,t-1} = Q_{rR,t-1}^{-1}$, while it is seldom pointed out that it is an approximation. The summary by $A_{rR,t-1}$ and $Q_{rR,t-1}$ is not exhaustive, but can still be included as a prior during the robust fitting at every time step:

$$e_{rR,t}(A) = \frac{1}{2}\sum_{i=1}^{i=n_t} \phi((\frac{X_{i,t}^t A - y_{i,t}}{\sigma})^2) + \frac{1}{2}(A - A_{rR,t-1})^t Q_{rR,t-1}(A - A_{rR,t-1})$$

Thus, $e_{rR,t}$ can be minimized by following the same approach as in Section 1.6.2 and 3.4.1. If for instance Cipra's approximate is used, the new approximate inverse covariance matrix at time step t is:

$$Q_{rR,t,Cipra} = \frac{1}{\sigma^2}\sum_{i=1}^{i=n_t} (\lambda_{i,t} X_{i,t} X_{i,t}^t) + Q_{rR,t-1,Cipra} \tag{3.8}$$

where $w_{i,t} = w_{i,j,t}$ with j the last iteration when the minimum of e_{rR} is reached. Similarly, other approximations can be derived from the approximate covariance matrices described in 3.3.

Finally, the recursive and robust algorithm consists of the following steps:

1. Initialize $Q_{rR,0}$ and $A_{rR,0}$ to zero or to a prior, and set t=1,
2. Initialize $A_{0,t} = A_{rR,t-1}$, and set $j = 1$,
3. For all indexes i ($1 \leq i \leq n_t$), compute the auxiliary variable $w_{i,j,t} = (\frac{X_{i,t}^t A_{j-1,t} - y_{i,t}}{\sigma})^2$, and the weights $\lambda_{i,j,t} = \phi(w_{i,j,t})$.
4. Solve the linear system

$$(\sum_{i=1}^{i=n_t} (\lambda_{i,j,t} X_{i,t} X_{i,t}^t) + Q_{rR,t-1}) A_{j,t} = \sum_{i=1}^{i=n_t} (\lambda_{i,j,t} X_i y_i) + Q_{rR,t-1} A_{rR,t-1}$$

5. If $\|A_{j,t} - A_{j-1,t}\| > \varepsilon$, increment j, and go to 3, else continue,
6. $A_{rR,t} = A_{j,t}$ and its approximate inverse covariance matrix $Q_{rR,t}$ is given by (3.8) or similar. If a new data set is available, increment t, and go to 2

In the recursive context, it is clear that a better estimate of the covariance matrix leads to better recursive estimators. In particular, if the covariance matrix is under-estimated with respect to the true covariance matrix, information about the past will be gradually lost. On the contrary, if the covariance matrix is over-estimated, the impact of the most recent data is always diminished.

3.5 Robust Kalman

Kalman filtering is a stochastic, recursive estimator, which estimates the state of a system based on the knowledge of the system input, the measurement of the system output, a model of the link between input and output, and state model.

We can identify state A_t at time t with $A_{rLS,t}$ or $A_{rR,t}$, depending on the pdf of the measurement noise. As in Section 3.2, we introduce $Y_t = (y_{i,t})_{1 \leq i \leq n_t}$, which is the so-called measurement vector, and $X_t = (X_{i,t})_{1 \leq i \leq n}$, the measurement matrix. The link between measurements and state can thus be written as $Y_t = X_t A_t + B$ where B is a vector of iid, centered measurement noises. This equation is the so-called measurement equation.

Compared to the recursive least squares, discrete Kalman filtering consists in assuming linear dynamics for the state model. More precisely, we assume $A_t = U_t A_{t-1} + V_t + u$ where u is a centered iid Gaussian model noise. This last equation is the so-called model, or state-transition, equation. As a summary, the Kalman model is:

$$\begin{cases} A_t &= U_t A_{t-1} + V_t + u \\ Y_t &= X_t A_t + v \end{cases} \tag{3.9}$$

When v is Gaussian, (3.9) yields the usual Kalman filter (subscript K). When v is non Gaussian, (3.9) models the robust to non-Gaussian measurement Kalman, or robust Kalman for short (subscript RK). The steady-state case we dealt with in the previous section, is a particular case of (3.9), where the first equation is deterministic and reduced to $A_{t+1} = A_t$.

In the dynamic case with v Gaussian, the prior on A is not A_{t-1} but the prediction $\hat{A}_{K,t} = U_t A_{K,t-1} + V_t$, given by the model equation. Using the model equation, the covariance matrix of the prediction $\hat{A}_{K,t}$ is derived as $\hat{C}_{K,t} = U_t C_{K,t-1} U_t^t + \Sigma$, where Σ is the covariance matrix of the Gaussian model noise u. Thus the inverse covariance matrix of the prediction $\hat{Q}_{K,t} = C_{K,t}^{-1}$ using the matrix lemma, is:

$$\hat{Q}_{K,t} = \Sigma^{-1} - \Sigma^{-1} U_t (U_t^t \Sigma^{-1} U_t + Q_{K,t-1})^{-1} U_t^t \Sigma^{-1} \tag{3.10}$$

This last equation is interesting in the context of real-time applications, since it involves only one matrix inverse at every time t. As in (3.5), the prediction is used as a Gaussian prior on A. The associated error, to be compared with (3.5), is now:

$$e_{K,t}(A) = \frac{1}{2} \sum_{i=1}^{i=n_t} \frac{1}{\sigma^2} (X_{i,t}^t A - y_{i,t})^2 + \frac{1}{2} (A - \hat{A}_{K,t})^t \hat{Q}_{K,t} (A - \hat{A}_{K,t})$$

The recursive equations of the Kalman filter are obtained by derivations from $e_{K,t}$. When $\hat{Q}_{K,t}$ is computed, only one linear system has to be solved at every t.

How does this method extend to the robust case? As for recursive least squares, an exact solution of the robust Kalman is generally not achievable. The two levels of approximations must be performed. Like in Section 3.4, we assume that $A_{RK,t-1}$ is approximately Gaussian, and its inverse covariance matrix is given by one of the approximations of Section 3.3. As a consequence, the associated error is:

$$e_{RK,t}(A) = \frac{1}{2}\sum_{i=1}^{i=n_t} \phi((\frac{X_{i,t}^t A - y_{i,t}}{\sigma})^2) + \frac{1}{2}(A - \hat{A}_{RK,t})^t \hat{Q}_{RK,t}(A - \hat{A}_{RK,t})$$

In the robust Kalman, Cipra's approximation (3.8) translates as:

$$Q_{RK,t,Cipra} = \frac{1}{\sigma^2}\sum_{i=1}^{i=n_t} (\lambda_{i,t} X_{i,t} X_{i,t}^t) + \hat{Q}_{RK,t,Cipra} \tag{3.11}$$

Other approximate inverse covariance matrix can be derived using approximates in Section 3.3.

Finally, the robust Kalman algorithm consists of the following steps:

1. Initialize $Q_{RK,0}$ and $A_{RK,0}$ to zero or to a prior, and set t=1,
2. Compute the predicted solution $\hat{A}_{RK,t} = U_t A_{RK,t-1} + V_t$, and its covariance matrix $\hat{Q}_{RK,t}$ using (3.10),
3. Initialize $A_{0,t} = \hat{A}_{RK,t}$, and set $j = 1$,
4. For all indexes i ($1 \leq i \leq n_t$), compute the auxiliary variable $w_{i,j,t} = (\frac{X_{i,t}^t A_{j-1,t} - y_{i,t}}{\sigma})^2$, and the weights $\lambda_{i,j,t} = \phi'(w_{i,j,t})$.
5. Solve the linear system
$$(\sum_{i=1}^{i=n_t} \lambda_{i,j,t} X_{i,t} X_{i,t}^t + \hat{Q}_{RK,t}) A_{j,t} = \sum_{i=1}^{i=n_t} \lambda_{i,j,t} X_i y_i + \hat{Q}_{RK,t} \hat{A}_{RK,t}$$
6. If $\|A_{j,t} - A_{j-1,t}\| > \varepsilon$, increment j, and go to 4, else continue,
7. $A_{RK,t} = A_{j,t}$ and its approximate inverse covariance matrix $Q_{RK,t}$ is given by (3.11) or similar. If a new data set is available, increment t, and go to 2.

Note that in [8][33], one single weighted least squares iteration is performed at each time step. We believe for each iteration one should achieve convergence in the approximation done in steps 4-6. Moreover the weights in [33] are binary. This corresponds to a truncated Gaussian pdf, violating **H0**.

As a conclusion, the Lagrange reformulation of half-quadratic approach of robust fitting allows us to have new insight into the reasons why robust Kalman filtering provides approximate estimates. Robust Kalman is not exact because: the amount of past data cannot be reduced

Figure 3.4: *Fitting and associated uncertainty curves along a sequence of images where the left-most lane-marking is occluded*

without loss of information, and the covariance matrix of the predicted state is an approximation. Contrary to [8][33], this formulation also suggests that it is important to iteratively search for the best solution A_t at every time step.

Figure 3.4 shows a sequence of images where the left-most lane-marking is occluded by a vehicle. Despite the occlusion, the lane-marking is tracked along the sequence. Notice how uncertainty increases during occultation, and decreases after.

3.6 Conclusion

The advantage of considering object detection as a statistical estimation problem is that the inverse covariance matrix naturally provides a measure of the confidence that can be associated with the detection result. Due to the non linearity in the non Gaussian assumption, an exact definition of the covariance matrix seems intractable in the robust case and approximations are needed. Moreover, it seems that the independence of noise is not satisfied in practice. We thus propose a correction factor to take into account the data correlation between image lines. This correction factor is used with success in lane detection system.

Moreover, we have reviewed the problem of making Kalman filtering robust to outliers. We have shown that, as soon as non-Gaussian likelihoods are involved, two levels of approximation are needed. First, there is no obvious closed-form expressions for the covariance matrix. An accurate covariance matrix is very important to tackle the problem of missing data on a long sequence of images. Second, the pdf of previous estimates must be considered as Gaussian. In existing algorithms, only one iteration is performed at each time step to obtain the robust estimate. We believe it is better to let the iterative re-weighted least squares algorithm converge.

It would be better to derive an integrated and more consistent framework without these two approximations. It is still an open question.

Chapter 4

Simultaneous Robust Fitting of Multiple Curves

4.1 Introduction

In this chapter, we propose a method for robustly recovering several instances of a curve model from a single noisy data set with severe perturbations (outliers). It is based on an extension of Chapter 1, in which M-estimators were revisited in Lagrangian formalism which lead to a new derivation and convergence proof of the well-known Iterative Reweighted Least Squares (IRLS) algorithm. Following the same approach based on the Lagrangian framework, we derive, in a natural way, a deterministic, alternate minimization algorithm, for Simultaneous Multiple Robust Fitting (SMRF) which extends the IRLS algorithm. Compared to the IRLS, the SMRF features an extra probability ratio, which is classical in clustering algorithms, in the expression of the weights. Potential numerical issues are tackled by banning zero probabilities in the computation of the weights and by introducing a Gaussian prior on the curves coefficients. Such a prior is, moreover, well-suited to sequential image processing and provides control on the curves. Applications to camera calibration and lane-markings tracking illustrate the effectiveness of the SMRF algorithm. In particular, it outperforms classical Gaussian mixture model algorithms in the presence of outliers.

The chapter is organized as follows. In Section 4.2, we present the robust multiple curves estimation problem and introduce our algorithmic strategy. The resulting algorithm is given in Section 4.3, while technical details on its derivation and convergence proof are given in Appendix 6.3. In Section 4.4, connections are made with other approaches in the domain. Finally, we apply the algorithm to road tracking and to camera calibration with respect to the road, in Section 4.5. Most of this chapter was first publish in [38, 37].

4.2 Multiple robust Maximum Likelihood Estimation (MLE)

In this section, we model the problem of simultaneously fitting m curves in a robust way. m is a priori fixed. Each individual curve is explicitly described by a vector parameter $\tilde{A}_j$, $1 \leq j \leq m$. The observations, y, are given by a linear generative model:

$$y = X(x)^t \tilde{A}_j + b \tag{4.1}$$

where (x,y) are the image coordinates of a data point, $\tilde{A}_j = (\tilde{a}_{ij})_{0\leq i\leq d}$ is the vector of curve parameters and $X(x) = (f_i(x))_{0\leq i\leq d}$ is the vector of basis functions at the image coordinate x, which will be denoted as X for the sake of simplicity. These vectors are of size $d+1$. Example basis functions were given in Section 1.5. Note that we consider the *fixed design* case, *i.e.* in (4.1), x is assumed non-random. In that case, it is shown, as explained in Chapter 2, that certain M-estimators attain the maximum possible breakdown point of approximately 50%. In all that follows, the measurement noise b is assumed independent and identically distributed (iid) and centered. In order to render the estimates robust to non-Gaussian noise (outliers), we formulate the noise distribution as:

$$p_s(b) \propto \frac{1}{s} e^{-\frac{1}{2}\phi((\frac{b}{s})^2)} \tag{4.2}$$

where $\propto$ denotes the equality up to a factor, and s is the scale of the pdf. The scale parameter, s, controls the distance from which noisy measurements have a good chance of being considered as outliers. As explained in Chapter 1, $\phi(t)$ must fulfill the **H0**, **H1** and **H2** hypotheses defined in Section 1.6.2.

Our goal is to simultaneously estimate the m curve parameter vectors $A_{j=1,\cdots,m}$ from the whole set of n data points (x_i,y_i), $i = 1,\cdots,n$. The probability of a measurement point (x_i,y_i), given the m curves is the sum of the probabilities over each curve:

$$p_i((x_i,y_i)|A_{j=1,\cdots,m}) \propto \frac{1}{s}\sum_{j=1}^{j=m} e^{-\frac{1}{2}\phi((\frac{X_i^t A_j - y_i}{s})^2)}.$$

Concatenating all curve parameters into a single vector $A = (A_j), j = 1,\cdots,m$ of size $m(d+1)$, we can write the probability of the whole set of points as the product of the individual probabilities:

$$p((x_i,y_i)_{i=1,\cdots,n}|A) \propto \frac{1}{s^n}\prod_{i=1}^{i=n}\sum_{j=1}^{j=m} e^{-\frac{1}{2}\phi((\frac{X_i^t A_j - y_i}{s})^2)} \tag{4.3}$$

Maximizing the likelihood $p((x_i,y_i)_{i=1,\cdots,n}|A)$ is equivalent to minimizing the negative of its logarithm:

$$e_{MLE}(A) = \sum_{i=1}^{i=n} -\ln(\sum_{j=1}^{j=m} e^{-\frac{1}{2}\phi((\frac{X_i^t A_j - y_i}{s})^2)}) + n\ln(s) \tag{4.4}$$

Using the same trick as the one described in [40] for robust fitting of a single curve, we introduce the auxiliary variables $w_{ij} = (\frac{X_i^t A_j - y_i}{s})^2$, as explained in Appendix 6.3. We then rewrite the value $e_{MLE}(A)$ as the value achieved at the unique saddle point of the following Lagrange function:

$$L_R = \sum_{i=1}^{i=n}\sum_{j=1}^{j=m}\frac{1}{2}\lambda_{ij}(w_{ij} - (\frac{X_i^t A_j - y_i}{s})^2) + \sum_{i=1}^{i=n}\ln(\sum_{j=1}^{j=m} e^{-\frac{1}{2}\phi(w_{ij})}) - n\ln(s) \tag{4.5}$$

Then, the algorithm obtained by alternated minimizations of the dual function w.r.t. λ_{ij} and A is globally convergent, towards a local minimum of $e_{MLE}(A)$, as shown in Appendix 6.3.

4.3 Simultaneous Multiple Robust Fitting Algorithm

As detailed in Appendix 6.3, minimizing (4.5) leads to alternate between the three sets of equations:

$$w_{ij} = (\frac{X_i^t A_j - y_i}{s})^2, 1 \leq i \leq n, 1 \leq j \leq m, \tag{4.6}$$

$$\lambda_{ij} = \frac{e^{-\frac{1}{2}\phi(w_{ij})}}{\sum_{k=1}^{k=m} e^{-\frac{1}{2}\phi(w_{ik})}} \phi'(w_{ij}), 1 \leq i \leq n, 1 \leq j \leq m, \tag{4.7}$$

$$(\sum_{i=1}^{i=n} \lambda_{ij} X_i X_i^t) A_j = \sum_{i=1}^{i=n} \lambda_{ij} y_i X_i, 1 \leq j \leq m \tag{4.8}$$

In practice, some care must be taken, to avoid numerical problems and singularities. First, it is important that the denominator in (4.7) be numerically non-zero, even for data points located far from all curves. Zero probabilities are banned by adding a small value ε' (equal to the machine precision) to the exponential in the probability p_i of a measurement point. As a consequence, when a point with index i is far from all curves, $\phi'(w_{ij})$ is weighted by a constant factor, $1/m$, in (4.7).

Second, the linear system in (4.8) may be singular. To avoid this, it is necessary to enforce a Gaussian prior on the whole curves parameters with bias A^{pr} and covariance matrix C^{pr}. Note that the reason for introducing such a prior is not purely technical: it is indeed a very simple and useful way of taking into account application-specific a priori knowledge, as shown in Section 4.5.1 and 4.5.2. A default prior can be used. In practice, we found the following default prior very useful: the bias is zero, *i.e.* $A^{pr} = 0$, and the inverse covariance matrix is block diagonal where each diagonal block equals:

$$C^{pr-1} = r \int_{-1}^{1} X(x) X(x)^t dx \tag{4.9}$$

assuming that $[-1, 1]$ is the range where x varies. The regularization term $(A - A^{pr})^t C^{pr-1} (A - A^{pr})$ is added to (4.4) and (4.5). Therefore, the parameter r allows us to balance between the data fidelity term and the prior. The advantage of the default prior described above is that it is a model of the perturbations due to the image sampling.

Finally, the Simultaneous Multiple Robust Fitting algorithm (SMRF) is written as:

1. Initialize the number of curves m, the vector $A^0 = (A_j^0)$, $1 \leq j \leq m$, which gathers all curves parameters and set the iteration index to $k = 1$.

2. For all indexes i, $1 \leq i \leq n$, and j, $1 \leq j \leq m$, compute the auxiliary variables $w_{ij}^k = (\frac{X_i^t A_j^{k-1} - y_i}{s})^2$ and the weights $\lambda_{ij}^k = \frac{\varepsilon' + e^{-\frac{1}{2}\phi(w_{ij}^k)}}{m\varepsilon' + \sum_{j=1}^{j=m} e^{-\frac{1}{2}\phi(w_{ij}^k)}} \phi'(w_{ij}^k)$.

3. Solve the linear system:

$$\left[D + C^{pr-1}\right] A^k = \left[\sum_{i=1}^{i=n} \lambda_{i1}^k y_i X_i \vdots \sum_{i=1}^{i=n} \lambda_{im}^k y_i X_i\right] + C^{pr-1} A^{pr}.$$

4. If $\|A^k - A^{k-1}\| > \varepsilon$, increment k, and go to 2, else the solution is $A = A^k$.

In the above algorithm, D is the block-diagonal matrix whose m diagonal blocks are the matrices $\sum_{i=1}^{i=n} \lambda_{ij}^k X_i X_i^t$ of size $(d+1) \times (d+1)$, with $1 \leq j \leq m$. The prior covariance matrix C^{pr} is of size $m(d+1) \times m(d+1)$. The prior bias A^{pr} is a vector of size $m(d+1)$, as well as A and A^k. The complexity is $O(nm)$ for the step 2, and $O(m^2(d+1)^2)$ for the step 3 of the algorithm.

4.4 Connections with other approaches

The proposed algorithm bears similarities with previous works in the field of regression and clustering and we would like to highlight a few of them.

In the single curve case, $m = 1$, the SMRF algorithm is reduced to the Iterative Reweighted Least Squares presented in Chapter 1 and used extensively in M-estimators [20]. The SMRF and IRLS algorithms share very similar structures and it is important to notice that the main difference lies in the expression of the Lagrange multipliers λ_{ij}, see (4.7). Compared to the IRLS, the λ_{ij} are just weighted by an extra probability ratio, which is classical in clustering algorithms.

To make the connection with clustering clearer, let us substitute $Y = A_j + b$ to the generative model (4.1), where Y and A_j are vectors of same size and respectively denote a data points and a cluster centroid. The derivation described in Section 4.3 is still valid and the obtained algorithm turns out to be a clustering algorithm with m clusters, each cluster being represented by a vector, its centroid. The probability distribution of a cluster around its centroid is directly specified by the function ϕ. The obtained algorithm is thus able of modeling the Y_i's by a mixture of pdfs which are not necessarily Gaussian. The mixture problem is usually solved by the well-known Expectation-Maximization (EM) approach [11]. In the non-Gaussian case, the maximization (M) step implements robust estimation, which is an iterative process in itself. Hence, the resulting EM algorithm involves two nested loops, while the proposed algorithm features only one. An alternative to the EM approach is the Generalized EM (GEM) approach which consists in performing an approximate M-step: typically, only one iteration rather than the full maximization. The resulting algorithm in the robust case is identical to the one we derived in the Lagrangian framework (apart from the regularization of the singular cases). In our formalism however, no approximation is made in the derivation of the algorithm, in contrast with the GEM approach.

We also found that the SMRF algorithm is very close to an earlier work in the context of clustering [5]. However, to our knowledge, the latter was just introduced as an extra ad-hoc weighting over M-estimators without statistical interpretation and, moreover, singular configurations were not dealt with.

The SMRF algorithm is subject to the initialization problem since it only converges towards a local minimum. To tackle this problem, the Graduated Non Convexity approach (GNC) [2] is used to improve the chances of converging towards the global minimum. Details are given in Section 4.5.2.

4.5 Experimental Results

The proposed approach being based on a linear generative model, many applications could potentially be addressed using the SMRF algorithm. We now focus on two specific applications, namely simultaneous lane-markings tracking and camera calibration from a regular lattice of lines with geometric distortions.

4.5.1 Geometric Priors

As noticed in Section 4.3, the use of a Gaussian prior allows introducing useful application-specific knowledge. For example, using (4.9) for the diagonal blocks of the inverse prior covariance matrix, we take into account perturbations due to image sampling.

Tuning the diagonal elements of C^{pr} provides control on the curve degree. For polynomials, the diagonal components of the covariance matrix correspond to monomials of different degrees. The components of degree higher than one are thus set to smaller values than those of degree zero and one.

Geometric smooth constraints between curves can be enforced by using also non-zero off-diagonal blocks. In particular it is a way of maintaining parallelism between curves. As an illustration, to smoothly enforce parallelism between two lines $y = a_0 + a_1 x$ and $y = a'_0 + a'_1 x$, *i.e.* to keep a_1 and a'_1 as close as possible, the prior covariance matrix is obtained by rewriting $(a_1 - a'_1)^2$ in matrix notations:

$$\begin{bmatrix} a_0 \\ a_1 \\ a'_0 \\ a'_1 \end{bmatrix}^t \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a'_0 \\ a'_1 \end{bmatrix}$$

The above matrix, multiplied by an overall factor can be used as an inverse prior covariance C^{pr-1}. The factor controls the balance between the data fidelity term and the other priors. Other kinds of geometric smooth constraints can be handled in a similar way, such as intersection at a given point, or symmetric orientations. These geometric priors can be combined by adding the associated regularization term $(A - A^{pr})^t C^{pr-1}(A - A^{pr})$ to (4.4).

4.5.2 Lane-Markings Tracking

We shall now describe the application of the SMRF algorithm to the problem of tracking lane markings. In addition to the arguments that we reported in the previous section, another interesting feature of using a Gaussian prior is that the SMRF is naturally suitable for being included in a Kalman filter as explained in Chapter 3. However, this raises the question of the definition of the posterior covariance matrix of the estimate. The approximate posterior covariance matrix for the whole set of curve parameters, A, is simply built as a block-diagonal matrix made of the individual posterior covariance matrices for each curve, C_j. The correlation of the noise can be approximately handle by a correction factor as described in 3.3.4. This temporal prior can be easily combined with geometric priors to allow tracking of parallel curves for instance.

Figure 4.1 shows the three curves simultaneously fitted on the lane-marking centers (in green) and the associated uncertainty curves of the horizontal position of each fitted curve

Figure 4.1: *Detected lane-markings (in green) and uncertainty about curve position (in red)*

($\pm\sqrt{X(x)^t C_j^{-1} X(x)}$, in red). Notice that the uncertainty is higher for the right and left sparse lane-markings than for the continuous axial one. Moreover, the higher the distance to the camera, the higher the uncertainty, since the curve gets closer to possible outliers. In all these experiments, and following ones, the parameters used for the noise model are $\alpha = 0.1$ and $s = 4$.

For the tracking itself, we experimented both separate Kalman filters on individual curves, and a simultaneous Kalman filter. The former can be seen as a particular case of the latter, in which the inverse prior covariance matrix C^{pr} is block-diagonal so the linear system of size $m(d+1)$ in the SMRF algorithm can be decomposed as m linear independent systems of size $d+1$. Figure 4.2 compares the results obtained with separate and simultaneous Kalman filters. Notice how the parallelism between curves is better preserved within the simultaneous Kalman filter, thanks to an adequate choice of the off-diagonal blocks of C^{pr}.

Figure 4.3 illustrates the ability of the SMRF-based Kalman filter to fit and track several curves in an image sequence. In that case, three lane-markings are simultaneously fitted and correctly tracked, even though the vehicle performs several lane changes during the 150-image sequence. Notice that, while Kalman filtering can incorporate a dynamic model of the vehicle,

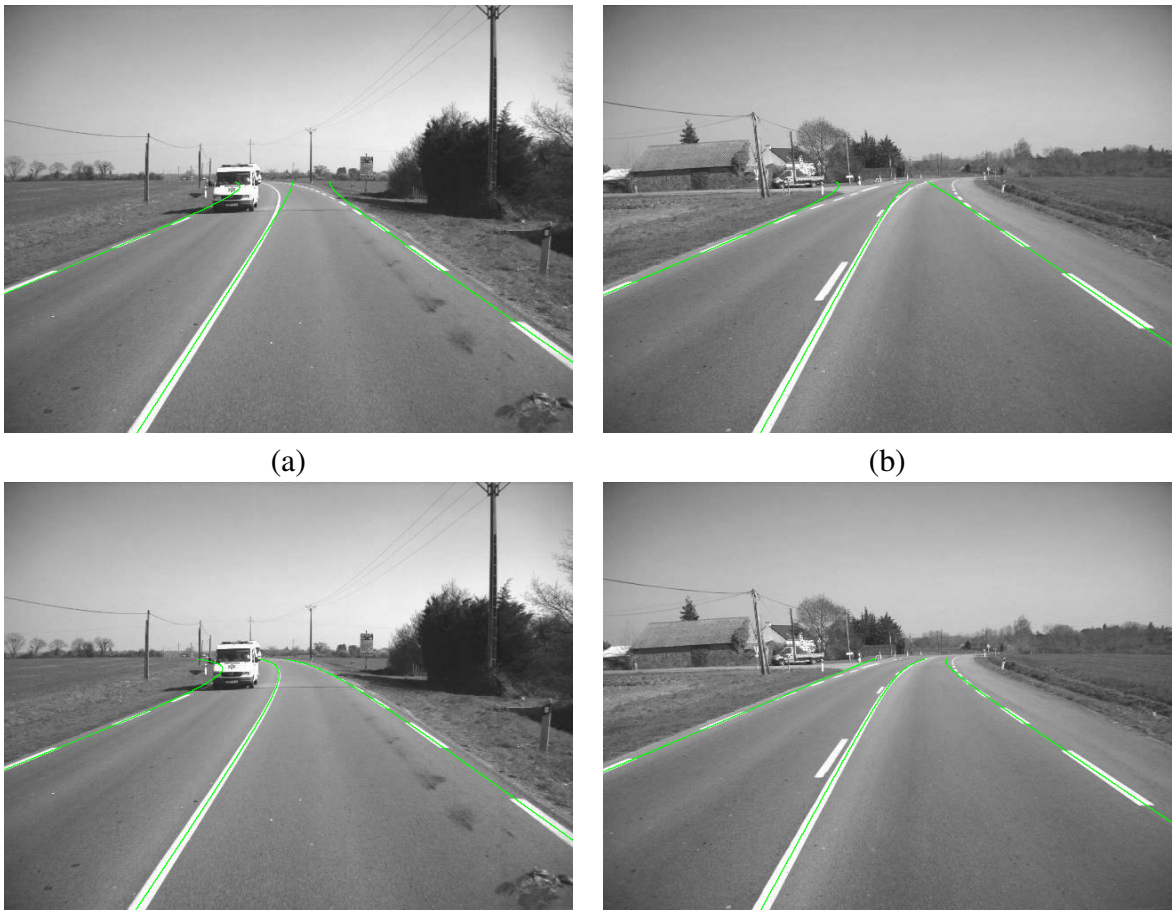


Figure 4.2: *Two images extracted from a sequence of 240 images processed with, on (a)(b), separate Kalman filters and, on (c)(d), simultaneous Kalman filter. The three detected lane-markings of degree two are in green*

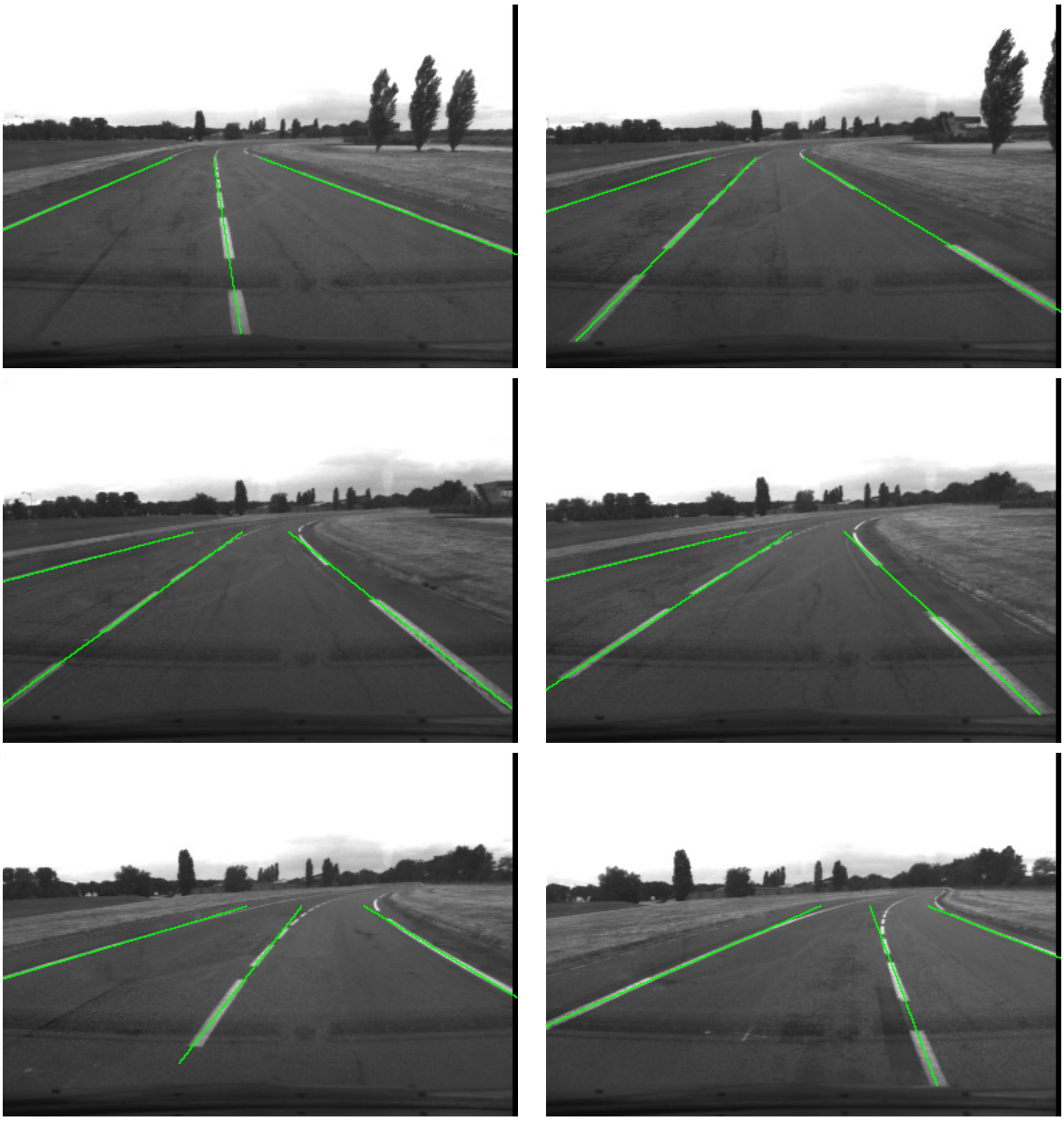


Figure 4.3: *Six of a 150-image sequence, featuring lane changes. Green lines show the three fitted lane-markings centers*

we only used a static model in these experiments, since only the images were available. We observed that it is better to initialize the SMRF algorithm with the parameters resulting from the fitting on the previous image, rather than with the filtered parameters: filtering indeed introduces a delay in the case of fast displacements or variations of the tracked curves.

Moreover, we obtained interesting results on difficult road sequences. For instance, Figure 4.4 shows a short sequence of poor quality images, due to rain. The left lane-marking is mostly hidden on two consecutive images. Thanks to the simultaneous Kalman filter, the SMRF algorithm is able to interpolate correctly the hidden lane-marking.

4.5.3 Camera Calibration

We now present another application of the SMRF algorithm, in the context of camera calibration. The goal is to accurately estimate the position and orientation of the camera with respect to the road and to its intrinsic parameters. A calibration pattern made of two sets of perpendicular lines painted on the road is observed by a camera mounted on a vehicle, as shown in Figure 4.5(a). The SMRF algorithm can be used to provide accurate data to the calibration algorithm by estimating the grid intersections. Even though the markings are clearly visible in the image, some of them are quite short, and there are outliers due to the presence of water puddles. Figure 4.5(b) shows the extracted lane-marking centers. When a Gaussian mixture model is used, the obtained fit is severely troubled by the outliers, as displayed in Figure 4.5(d), even though the curves are initialized very close to the expected solution, see Figure 4.5(c).

On the contrary, with the same extracted lane-marking centers, the SMRF algorithm, with noise model parameters $\alpha = 0.1$ and $s = 4$, leads to nice results, as shown in Figure 4.6(b) for the vertical lines, and in Figure 4.6(c) for the horizontal curves. 11 different lines were obtained for the vertical markings, and 12 different second degree polynomials were obtained for the horizontal markings. Figure 4.6(d) shows the two sets of curves superimposed.

We also use these calibration images to investigate the issue of initialization. We typically take a number of curve prototypes higher than the real number of curves in the image, see e.g. Figure 4.6(a). We observed that the extra prototypes may be either fitted on outliers groups or identical to another fitted prototype (e.g. in Figure 4.6(b), two resulting curves are identical). Detecting identical curves is easy, for instance by performing a Bayesian recognition test on every pair (A_j, A_k), *i.e.* comparing $(A_j - A_k)^t C_j^{-1} (A_j - A_k)$ to a small threshold, where C_j is the posterior covariance matrix of the curve A_j. To detect prototypes fitted to only a few points such as outliers, we exploit the uncertainty measure provided by the posterior covariance matrix and simply threshold $-ln(det(C_j))$. Finally, notice that the SMRF algorithm does not require to be initialized very close to the expected solution, as illustrated by Figure 4.6(a)(b).

4.6 Conclusion

In the continuing quest for achieving robustness in detection and tracking curves in images, this chapter makes two contributions. The first one is the derivation, in a MLE approach and using Kuhn and Tucker's classical theorem, of the so-called SMRF algorithm. This algorithm extends mixture model algorithm, such as the one derived using EM, to robust curve fitting. It is also an extended version of the IRLS, in which the weights incorporate an extra probability ratio. The

Figure 4.4: *Fitting in adverse conditions: in this excerpt, the left lane-marking is mostly hidden on two successive images*

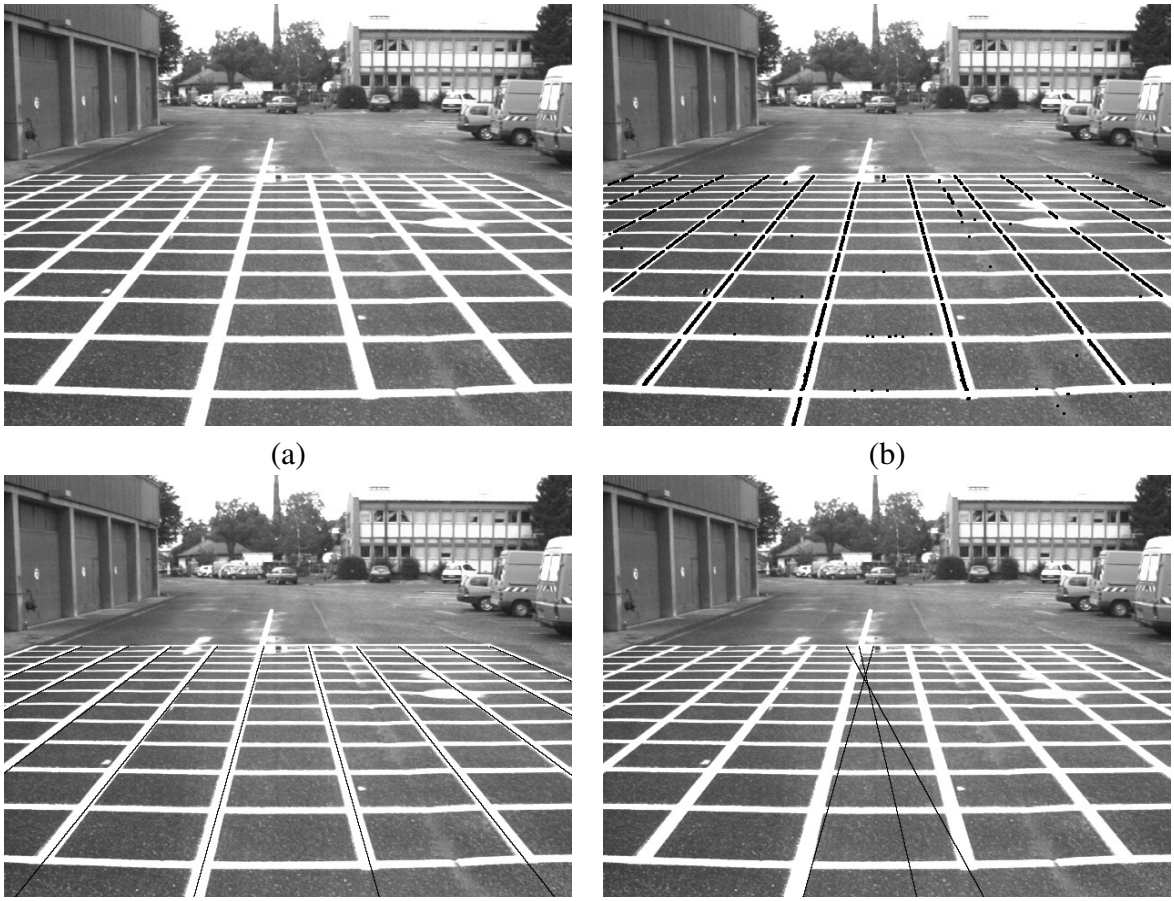


Figure 4.5: *(a) Original image of the calibration grid.*
(b) Extracted lane-marking centers (outliers are due to puddles).
(c) 10 initial lines for the fitting on the vertical markings.
(d) Fitted lines on the vertical markings under Gaussian noise assumption

second contribution is the regularization of the SMRF algorithm by introducing Gaussian priors on curve parameters and the handling of potential numerical issues by banning zero probabilities in the computation of weights. From our experiments, banning zero probabilities seems to have important positive consequences in pushing the curves to spread out all the data, and thus in providing improved robustness to the initialization, as shown in the context of camera calibration. The introduction of the Gaussian prior is also beneficial in particular in the context of image sequence processing, as illustrated with an application of simultaneous lane-markings tracking on-board a vehicle in adverse conditions. The approach being based on a linear generative model, it is quite generic and we believe that it can be used with benefits in many other fields, such as clustering or appearance modeling.

Finally, notice that the approach proposed in this chapter lead to an operational, real-time guidance system, which was successfully tested within the context of the ARCOS 2004 project. Moreover, a multiple lines detection algorithm is now routinely used for calibrating the MLPC IRCAN (French acronym for Road Imagery using Numeric CAmeras) systems.

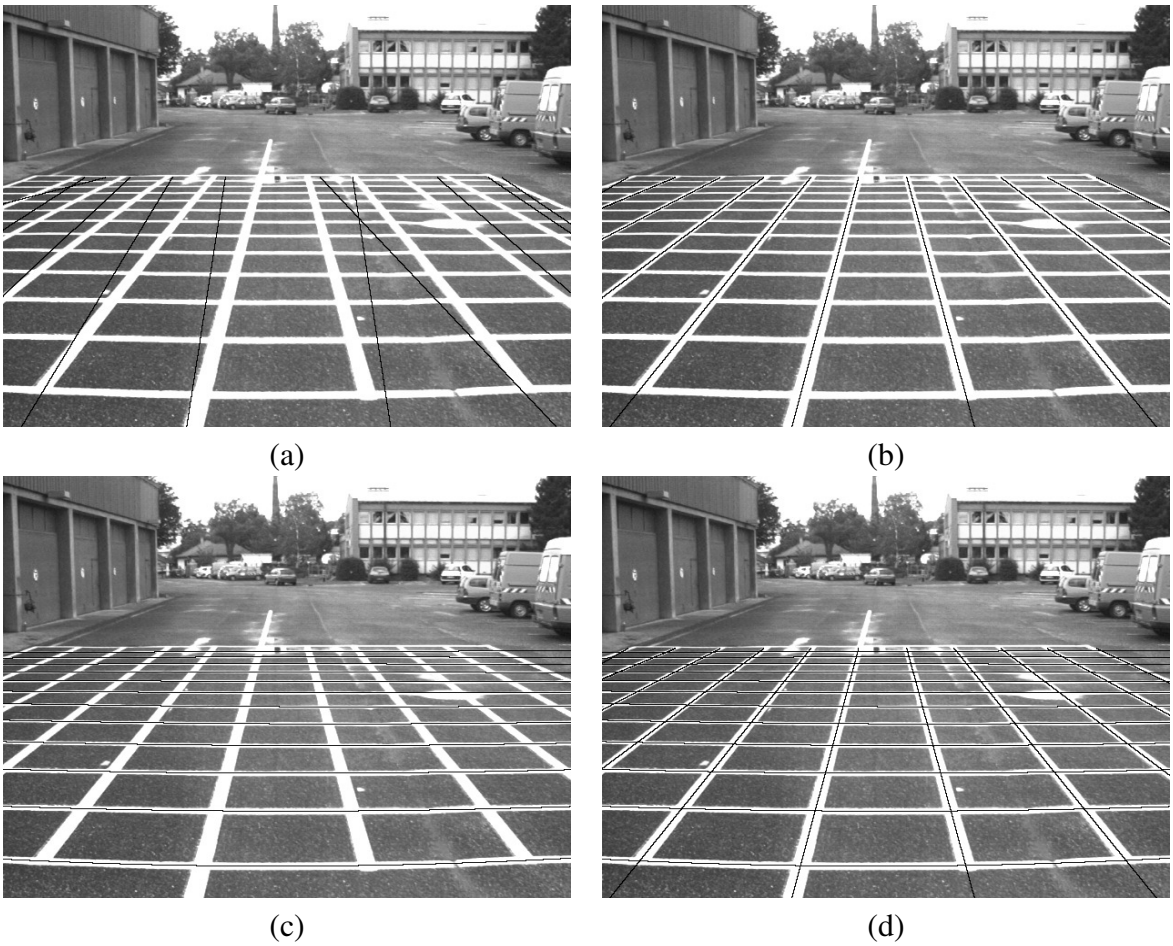


Figure 4.6: *(a) 12 initial lines for the fitting on the vertical markings.*
(b) The fitting yields 11 different lines.
(c) 12 second degree polynomials fitted on the horizontal markings.
(d) Results of the horizontal and vertical fitting superimposed

Chapter 5

Conclusions

Automatic road marking detection from real images of road scenes requires both robustness and ability to evaluate the detection results. Our approach considers detection as a robust regression problem over polynomial bases. We propose a framework based on the half-quadratic approach of statistical estimation, which allows to derive algorithms which are:

- *robust* thanks to the Lagrangian reformulation of the half-quadratic approach,
- *fast* even if iterative,
- *convergent* towards a local minimum close to the global one, thanks to the Graduated Non-Convexity strategy,
- able to provide a *complete evaluation* of each detection result, thanks to the approximate covariance matrix,
- able to tackle one or *several markings* simultaneously,
- suitable for embedding in a Kalman filter for *tracking* along image sequences.

Experimental results assess the validity of this approach that was successfully used for both lane following and camera calibration.

Of course several points can still be improved. In particular, a few extra questions needs to be further investigated:

- More robustness may be achieved using region fitting rather than edge fitting. Preliminary results on region fitting using the half-quadratic approach were obtained in [1], also using the Lagrange reformulation, in the context of road pavement detection.
- When stereo cameras are available, the vertical profile of the road can be estimated, see [41, 7], and substituted with advantages to the planar road assumption, which is not always accurate for long distances.
- The derivation of the exact covariance matrix of the robust estimate, taking into account the correlation of the data, remains an open theoretical question. Answering this question should improve the results of lane marking detection and is also of main importance for many other applications where robust regression is involved.

Acknowledgments

This work was partly funded by the Région Île de France. The authors are grateful to the LIVIC's technical team, and in particular to Benoît Lusetti, for their help in the experiments.

Chapter 6

Appendix

6.1 Constrained Optimization in the IRLS

To prove the local convergence of the IRLS in section 1.6.3, we now focus on the minimization of $E(A,W)$ w.r.t. W only subject to the n constraints $h_i(A,W) \leq 0$, w.r.t. W, for any A. This problem is well-posed because it is a minimization of a convex function subject to convex constraints. Therefore using Kuhn and Tucker's classical theorem [29], if a solution exists, the minimization of $E(A,W)$ w.r.t. W is equivalent to searching from the unique saddle point of the Lagrange function of the problem:

$$L_R(A,W,\lambda_i) = \frac{1}{2}\sum_{i=1}^{i=n} -\phi(w_i) + \lambda_i(w_i - (\frac{X_i^t A - y_i}{s})^2)$$

where λ_i are Kuhn and Tucker multipliers ($\lambda_i \geq 0$). More formally, we have proved for any A:

$$-e_R(A) = \min_{w_i} \max_{\lambda_i} L_R(A,W,\lambda_i) \tag{6.1}$$

Notice that the Lagrange function L_R is now quadratic w.r.t. A, unlike the original error e_R. Using the saddle point property, we can change the order of variables w_i and λ_i in (6.1). $L_R(A,W,\lambda_i)$ being convex w.r.t W, it is equivalent to searching for a minimum of $L_R(A,W,\lambda_i)$ w.r.t. W and to having its first derivative zero. Thus, we deduce:

$$\lambda_i = \phi'(w_i) \tag{6.2}$$

This last equation can be used with **H2** to substitute w_i in L_R and then to deduce that the original problem is equivalent to the following minimization:

$$\min_A e_R(A) = \min_{A,\lambda_i} -L_R(A, {\phi'}^{-1}(\lambda_i), \lambda_i)$$

The *dual function* $\mathcal{E}(A,\lambda_i) = -L_R(A, {\phi'}^{-1}(\lambda_i), \lambda_i)$ is convex w.r.t. A. $\mathcal{E}$ is also convex w.r.t. λ_i (Indeed, $\frac{\partial^2 \mathcal{E}}{\partial \lambda_i^2} = -\frac{1}{\phi''({\phi'}^{-1}(\lambda_i))}$). The robust algorithm, like the classical one used in half-quadratic and M-estimators approaches, consists in minimizing $\mathcal{E}$ w.r.t. A and λ_i alternately. It can be shown that such an algorithm always strictly decreases the dual function if the current point

is not a stationary point (*i.e.* a point where the first derivatives are all zero) of the dual function [27]. The problem of stationary points is easy to solve by checking the positiveness of the Hessian matrix of $\mathcal{E}(A,\lambda_i)$. If this matrix is not positive, we perturb the solution so that it converges to a local minimum. Using the previous Lagrange function, this proves that the algorithm is globally convergent, *i.e.*, it converges toward a local minimum of $e_R(A)$ for all initial A_0s which are not a maximum or a saddle point of $e_R(A)$. A similar result has been shown in [31, 10], considering the stationary point of the cost function $e_R(A)$ instead of the dual function.

With Lagrangian formulation, the uniqueness of the solution can be shown when $\mathcal{E}$ is convex w.r.t. the couple (A,λ_i).

6.2 Covariance Matrix Derivation

Let us justify (3.3) in Section 3.3. Since, as shown below, the estimator is unbiased, the covariance matrix of the estimated parameter vector A is defined as the expectation $E[(A-\tilde{A})(A-\tilde{A})^t]$, where $\tilde{A}$ is the true parameter vector. We rewrite step 3 of the robust algorithm as $O_1A = B$ with $B = \sum_{i=1}^{i=n}\lambda_i y_i X_i$. For the true parameters, we have a similar equation $\tilde{O}_1\tilde{A} = \tilde{B}$ where $\tilde{B} = \sum_{i=1}^{i=n}\tilde{\lambda}_i\tilde{y}_i X_i$. In the following, we forget about the fact that λ_i is a random variable and thus $\tilde{\lambda}_i = \lambda_i$. First, this implies $\tilde{O}_1 = O_1$ or $A-\tilde{A} = O_1^{-1}(B-\tilde{B})$. Second, since $y_i = \tilde{y}_i + b_i$, we deduce $B-\tilde{B} = \sum_{i=1}^{i=n}\lambda_i b_i X_i$. Since the noise is centered this confirms that $E[A-\tilde{A}] = 0$.The estimator is unbiased. Moreover, b_i are independent random variables, thus $E[(B-\tilde{B})(B-\tilde{B})^t] = \sum_{i=1}^{i=n}\lambda_i^2 E[b_i^2]X_iX_i^t = E[b^2]O_2$. As a consequence, the covariance matrix of A can be approximated by $O_1^{-1}E[(B-\tilde{B})(B-\tilde{B})^t]O_1^{-1}$ or:

$$C_{ITC} = E[b^2]O_1^{-1}O_2O_1^{-1} \tag{6.3}$$

after substitution of the covariance matrix of B. The question is now how to approximate the variance of the noise $E[b^2]$. The simplest idea is to compute it from the noise model, but this could lead to bad estimates when the noise model is not accurate enough. A better idea, leading to estimates with better behavior in the case of an inaccurate choice for the noise model, is to derive $E[b^2]$ from the data residuals b_i using (2.11). We now derive the relationship between $E[b^2]$ and $E[\hat{s}^2]$, always assuming that λ_i are not random. Equation (2.11) is rewritten as $\hat{s}^2 = \frac{1}{n}\sum_{i=1}^{i=n}\lambda_i(y_i - X_i^tA)^2$. By introducing the true parameter vector $\tilde{A}$, inside the sum, we get:

$$\hat{s}^2 = \frac{1}{n}\sum_{i=1}^{i=n}\lambda_i(b_i + X_i^t(\tilde{A}-A))^2 \tag{6.4}$$

To calculate $E[\hat{s}^2]$ as a function of $E[b^2]$, we expand the squared term in the previous equation. Using (6.3), we deduce:

$$E[\hat{s}^2] = \frac{1}{n}\sum_{i=1}^{i=n}(E[b^2]\lambda_i + E[b^2]\lambda_i X_i^t O_1^{-1}O_2O_1^{-1}X_i + 2\lambda_i X_i^t E[(\tilde{A}-A)b_i]) \tag{6.5}$$

Since b_i is centered, thus $E[(\tilde{A}-A)b_i] = -E[Ab_i]$ in (6.5). The estimated parameter vector obtained by fitting is $A = O_1^{-1}\sum_{k=1}^{k=n}\lambda_k X_k y_k = O_1^{-1}\sum_{i=1}^{k=n}\lambda_k X_k(X_k^t\tilde{A} + b_k)$. Therefore $E[Ab_i]$ can

be expanded as $E[Ab_i] = O_1^{-1}\sum_{k=1}^{k=n}\lambda_k X_k E[b_k b_i]$. The independence of b_i allows us to simplify the last equation in $E[Ab_i] = O_1^{-1}\lambda_i X_i E[b^2]$. As a consequence (6.5) is now:

$$E[\hat{s}^2] = \frac{E[b^2]}{n}(\sum_{i=1}^{i=n}\lambda_i + \sum_{i=1}^{i=n}\lambda_i X_i^t O_1^{-1} O_2 O_1^{-1} X_i - 2\sum_{i=1}^{i=n}\lambda_i^2 X_i^t O_1^{-1} X_i) \tag{6.6}$$

The last term in (6.6) can be rewritten as: $-2Trace(\sum_{i=1}^{i=n}\lambda_i^2 X_i^t O_1^{-1} X_i) = -2\sum_{i=1}^{i=n} Trace(\lambda_i^2 X_i^t O_1^{-1} X_i)$, so $-2\sum_{i=1}^{i=n} Trace(\lambda_i^2 X_i X_i^t O_1^{-1}) = -2Trace(O_2 O_1^{-1})$ using $Trace(AB) = Trace(BA)$. Similarly, the second term in (6.6) is also rewritten as $Trace(O_2 O_1^{-1})$. This implies $E[\hat{s}^2] = \frac{E[b^2]}{n}\left(\sum_{i=1}^{i=n}\lambda_i - Trace(O_2 O_1^{-1})\right)$. Finally, we approximate $E[\hat{s}^2]$ by $\hat{s}^2$, and using (6.4) with $A = \tilde{A}$, we deduce the following estimate of $E[b^2]$:

$$E[b^2] = \frac{\sum_{i=1}^{i=n}\lambda_i b_i^2}{\sum_{i=1}^{i=n}\lambda_i - Trace(O_2 O_1^{-1})} \tag{6.7}$$

By substitution of (6.7) in (6.3), we justify the proposed covariance matrix estimate (3.3).

6.3 Constrained Optimization in the SMRF

To prove the local convergence of the SMRF in section 4.2, we shall first rewrite the value $-e_{MLE}(A)$ for any given $A = (A_j), j = 1, \cdots, m$ as the value achieved at the minimum of a convex problem under convex constraints. This is obtained by introducing the auxiliary variables $w_{ij} = (\frac{X_i^t A_j - y_i}{s})^2$. Like in Appendix 6.1, this apparent complication is in fact valuable since it allows us to introduce Lagrange multipliers, and thus to decompose the original problem in simpler problems. The value $-e_{MLE}(A)$ can be seen as the minimum value, w.r.t. $W = (w_{ij})_{1\le i\le n, 1\le j\le m}$, of:

$$E(A,W) = \sum_{i=1}^{i=n} ln(\sum_{j=1}^{j=m} e^{-\frac{1}{2}\phi(w_{ij})})$$

subject to nm constraints $h_{ij}(A,W) = w_{ij} - (\frac{X_i^t A_j - y_i}{s})^2 \le 0$. This is proved by showing that the bound on each w_{ij} is always achieved. Indeed $E(A,W)$ is decreasing w.r.t. each w_{ij}, since its first derivative:

$$\frac{\partial E}{\partial w_{ij}} = -\frac{1}{2}\frac{e^{-\frac{1}{2}\phi(w_{ij})}}{\sum_{k=1}^{k=m} e^{-\frac{1}{2}\phi(w_{ik})}}\phi'(w_{ij})$$

is always negative, due to **H1**.

To prove the local convergence of the SMRF algorithm in Section 4.3, we now focus on the minimization of $E(A,W)$ w.r.t. W only, subject to the nm constraints $h_{ij}(A,W) \le 0$, w.r.t. W, for any A. We now introduce a classical result of convex analysis [4]: the function $g(Z) = ln(\sum_{j=1}^{j=m} e^{z_j})$ is convex. Due to **H1** and **H2**, $-\phi(w)$ is convex and decreasing. Therefore, $E(A,W)$ w.r.t. W is convex as a sum of functions g composed with $-\phi$, see [4]. As a consequence, the minimization of $E(A,W)$ w.r.t. W is well-posed because it is a minimization of a convex function subject to convex (linear) constraints. We are thus allowed to apply Kuhn and Tucker's classical theorem [29]: if a solution exists, the minimization of $E(A,W)$ w.r.t. W is

equivalent to searching from the unique saddle point of the Lagrange function of the problem:

$$\begin{aligned} L_R(A,W,\Lambda) &= \sum_{i=1}^{i=n} ln(\sum_{j=1}^{j=m} e^{-\frac{1}{2}\phi(w_{ij})}) \\ &+ \sum_{i=1}^{i=n}\sum_{j=1}^{m} \frac{1}{2}\lambda_{ij}(w_{ij} - (\frac{X_i^t A_j - y_i}{s})^2) \end{aligned}$$

where $\Lambda = (\lambda_{ij}), 1 \leq i \leq n, 1 \leq j \leq m$ are Kuhn and Tucker multipliers ($\lambda_{ij} \geq 0$). More formally, we have proved for any A:

$$-e_{MLE}(A) = \min_W \max_\Lambda L_R(A,W,\Lambda) \tag{6.8}$$

Notice that the Lagrange function L_R is quadratic w.r.t. A, unlike the original error e_{MLE}. Using the saddle point property, we can change the order of variables W and Λ in (6.8). We now introduce the *dual function* $\mathcal{E}(A,\Lambda) = \min_W L_R(A,W,\Lambda)$, and rewrite the original problem as the equivalent following problem:

$$\min_A e_{MLE}(A) = \min_{A,\Lambda} -\mathcal{E}(A,\Lambda)$$

The algorithm consists in minimizing $-\mathcal{E}(A,\Lambda)$ w.r.t. A and Λ alternately. $\min_\Lambda -\mathcal{E}(A,\Lambda)$ leads to Kuhn and Tucker's conditions:

$$\lambda_{ij} = \frac{e^{-\frac{1}{2}\phi(w_{ij})}}{\sum_{k=1}^{k=m} e^{-\frac{1}{2}\phi(w_{ik})}}\phi'(w_{ij}) \tag{6.9}$$

$$w_{ij} = (\frac{X_i^t A_j - y_i}{s})^2 \tag{6.10}$$

and $\min_{A_j} -\mathcal{E}(A,\Lambda)$ leads to:

$$(\sum_{i=1}^{i=n} \lambda_{ij} X_i X_i^t) A_j = \sum_{i=1}^{i=n} \lambda_{ij} y_i X_i,\ 1 \leq j \leq m \tag{6.11}$$

Using classical results, see e.g. [29], $-\mathcal{E}(A,\Lambda)$ is proved to be convex w.r.t. Λ. The dual function is clearly quadratic and convex w.r.t. A. As a consequence, this implies that such an algorithm always strictly decreases the dual function if the current point is not a stationary point (*i.e.* a point where the first derivatives are all zero) of the dual function [27]. The problem of stationary points is easy to solve by checking the positiveness of the Hessian matrix of $\mathcal{E}(A,\Lambda)$. If this matrix is not positive, we disturb the solution so that it converges to a local minimum. This proves that the algorithm is globally convergent, *i.e.*, it converges toward a local minimum of $e_{MLE}(A)$ for all initial A_0's which are neither a maximum nor a saddle point.

Bibliography

[1] E. Bigorgne and J.-P. Tarel. Backward segmentation and region fitting for geometrical visibility range estimation. In *Proceedings of Asian Conference on Computer Vision (ACCV'07)*, volume II, pages 817–826, Tokyo, Japan, 2007. http://perso.lcpc.fr/tarel.jean-philippe/accv07.html.

[2] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, MA, 1987.

[3] C. Bouman and K. Sauer. A generalized Gaussian image model for edge-preserving map estimation. *IEEE Trans. on Image Processing*, 2(3):296–310, 1993.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[5] N. A. Cambell. Mixture models and atypycal values. *Mathematical Geology*, 16(5):465–477, 1984.

[6] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, 6(2):298–311, 1997.

[7] P. Charbonnier, V. Muzet, P. Nicolle, N. Hautière, J.-P. Tarel, and D. Aubert. La stéréovision appliquée à l'analyse de scènes routières. *submited to Bulletin des laboratoires des Ponts et Chaussées*, march 2007. http://perso.lcpc.fr/tarel.jean-philippe/publis/bl07.html.

[8] T. Cipra and R. Romera. Robust kalman filtering and its application in time series analysis. *Kybernetika*, 27(6):481–494, 1991.

[9] R. Dahyot. *Appearance based road scene video analysis for management of the road network*. PhD thesis, University of Strasbourg I, 2001.

[10] A. H. Delaney and Y. Bresler. Globally convergent edge-preserving regularized reconstruction: An application to limited-angle tomography. *IEEE Transactions on Image Processing*, 7(2):204–221, February 1998.

[11] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38, 1977.

[12] E. D. Dickmanns and A. Zapp. A curvature-based scheme for improving road vehicle guidance by computer vision. In *Proceedings of SPIE Conference on Mobile Robots*, volume 727, pages 161–168, October 1986.

[13] F. Fleuret and D. Geman. Coarse-to-fine visual selection. *International Journal of Computer Vision*, 41(1/2):85–107, January 2001.

[14] D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(3):367–383, 1992.

[15] S. Geman and D. McClure. Statistical methods for tomographic image reconstruction. *Bull. Int. Stat. Inst.*, LII(4):5–21, 1987.

[16] F. Guichard and J.-P. Tarel. Curve extraction combining perceptual grouping and a kalman like fitting. In *IEEE International Conference on Computer Vision (ICCV'99)*, pages 1003–1008, Kerkyra, Greece, 1999. http://perso.lcpc.fr/tarel.jean-philippe/publis/iccv99.html.

[17] F.R. Hampel, P.J. Rousseeuw, E.M. Ronchetti, and W.A. Stahel. *Robust Statistics*. John Wiley and Sons, New York, New York, 1986.

[18] D. Hasler, L. Sbaiz, S. Ssstrunk, and M. Vetterli. Outlier modeling in image matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):301–315, March 2003.

[19] J. Huang and D. Mumford. Statistics of natural images and models. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition (CVPR'99)*, pages 1541–1547, Ft. Collins, CO, USA, 23-25 June 1999.

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

[21] P. J. Huber. Finite sample breakdown of m- and p- estimators. *Annal of Statistics*, 12:119–126, 1984.

[22] S. S. Ieng. *Méthodes robustes pour la détection et le suivi des marquages.* PhD thesis, Université Paris VI, France, 2004.

[23] S.-S. Ieng, J.-P. Tarel, and P. Charbonnier. Evaluation of robust fitting based detection. In *Proceedings of European Conference on Computer Vision (ECCV'04)*, pages 341–352, Prague, Czech Republic, 2004. http://perso.lcpc.fr/tarel.jean-philippe/publis/eccv04.html.

[24] S.-S. Ieng, J.-P. Tarel, and P. Charbonnier. Modeling non-Gaussian noise for robust image analysis. In *Proceedings of International Conference on Computer Vision Theory and Applications (VISAPP'07)*, pages 175–182, Barcelona, Spain, 2007. http://perso.lcpc.fr/tarel.jean-philippe/publis/visapp07a.html.

[25] S.-S. Ieng, J.-P. Tarel, and R. Labayrade. On the design of a single lane-markings detector regardless the on-board camera's position. In *Proceedings of IEEE Intelligent Vehicle Symposium (IV'2003)*, pages 564–569, Colombus, OH, USA, 2003. http://perso.lcpc.fr/tarel.jean-philippe/publis/iv03.html.

[26] Sio-Song Ieng, Jean-Philippe Tarel, and Pierre Charbonnier. Estimation robuste pour la détection et le suivi par caméra. *Traitement du Signal*, 21(3):205–226, 2004. http://perso.lcpc.fr/tarel.jean-philippe/publis/ts04.html.

[27] D. G. Luenberger. *Introduction to linear and nonlinear programming*. Addison Wesley, 1973.

[28] P. Meer, C. V. Stewart, and D. Tyler. Robust computer vision: An interdisciplinary challenge. *Computer Vision and Image Understanding: CVIU*, 78(1):1–7, 2000.

[29] M. Minoux. *Mathematical Programming: Theory and Algorithms*. Chichester: John Wiley and Sons, 1986.

[30] I. Mizera and C. Müller. Breakdown points and variation exponents of robust m-estimators in linear models. *The Annals of Statistics*, 27(4):1164–1177, 1999.

[31] J-C. Nosmas. Remarques sur un algorithme d'optimisation pour une classe de fonctionnelles sur rk. *C. R. Acad. Sci. Paris*, t.328(Série I)(Analyse numérique):1237–1240, 1999.

[32] J. M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348–365, 1995.

[33] R. Rao. Robust kalman filters for prediction, recognition, and learning. Technical Report TR645, University of Rochester, 1996.

[34] H. W. Sorenson. *Parameter Estimation*. Marcel Dekker, Inc., New York, New York, 1980.

[35] A. Srivastava, A. Lee, E. Simoncelli, and S. Zhu. On advances in statistical modeling of natural images. *Journal of Mathematical Imaging and Vision*, 18(1):17–33, 2003.

[36] J.-P. Tarel. *Perception par caméra des bords de route*. Collections Etudes et Recherches des Laboratoires des Ponts et Chaussées, LCPC, Janvier 2003. http://perso.lcpc.fr/tarel.jean-philippe/publis/rr02.html.

[37] J.-P. Tarel, P. Charbonnier, and S.-S. Ieng. Ajustement robuste et simultané de plusieurs courbes et applications au suivi de marquages routiers. In *Proceedings of Journées des Sciences de l'Ingénieur (JSI'06)*, Marne la Vallée, France, 2006. http://perso.lcpc.fr/tarel.jean-philippe/publis/jsi06a.html.

[38] J.-P. Tarel, P. Charbonnier, and S.-S. Ieng. Simultaneous robust fitting of multiple curves. In *Proceedings of International Conference on Computer Vision Theory and Applications (VISAPP'07)*, pages 175–182, Barcelona, Spain, 2007. http://perso.lcpc.fr/tarel.jean-philippe/publis/visapp07b.html.

[39] J.-P. Tarel and F. Guichard. Dynamic tracking of lane markings and boundaries in curved road. In *IEEE International Conference on Image Processing (ICIP'2000)*, volume I, pages 216–219, Vancouver, Canada, 2000. http://perso.lcpc.fr/tarel.jean-philippe/publis/icip00.html.

[40] J.-P. Tarel, S.-S. Ieng, and P. Charbonnier. Using robust estimation algorithms for tracking explicit curves. In *European Conference on Computer Vision (ECCV'02)*, volume 1, pages 492–507, Copenhagen, Danmark, 2002. http:///www-rocq.inria.fr/ tarel/eccv02.html.

[41] J.-P. Tarel, S.-S. Ieng, and P. Charbonnier. Accurate and robust image alignment for road profil reconstruction. In *Proceedings of IEEE International Conference on Image Processing (ICIP'07)*, volume V, pages 365–368, San Antonio, Texas, USA, 2007. http://perso.lcpc.fr/tarel.jean-philippe/icip07.html.

[42] T. Tasdizen, J.-P. Tarel, and D.B. Cooper. Improving the stability of algebraic curves for applications. *IEEE Transactions on Image Processing*, 9(3):405–416, March 2000. http://perso.lcpc.fr/tarel.jean-philippe/publis/ip00.html, Also as LEMS Tech. Report 176, Brown University.