Using Robust Estimation Algorithms for Tracking Explicit Curves

Jean-Philippe Tarel[1], Sio-Song Ieng[1], and Pierre Charbonnier[2]

[1] LIVIC (INRETS-LCPC), 13, Route de la Minière, 78000 Versailles, France.
tarel@lcpc.fr, ieng@inrets.fr
[2] LRPC, 11, Rue Jean Mentelin, BP 9, 67200 Strasbourg, France.
Pierre.Charbonnier@equipement.gouv.fr

Abstract. The context of this work is lateral vehicle control using a camera as a sensor. A natural tool for controlling a vehicle is recursive filtering. The well-known Kalman filtering theory relies on Gaussian assumptions on both the state and measure random variables. However, image processing algorithms yield measurements that, most of the time, are far from Gaussian, as experimentally shown on real data in our application. It is therefore necessary to make the approach more robust, leading to the so-called robust Kalman filtering. In this paper, we review this approach from a very global point of view, adopting a constrained least squares approach, which is very similar to the half-quadratic theory, and justifies the use of iterative reweighted least squares algorithms. A key issue in robust Kalman filtering is the choice of the prediction error covariance matrix. Unlike in the Gaussian case, its computation is not straightforward in the robust case, due to the nonlinearity of the involved expectation. We review the classical alternatives and propose new ones. A theoretical study of these approximations is out of the scope of this paper, however we do provide an experimental comparison on synthetic data perturbed with Cauchy-distributed noise.

1 Introduction

Automatic driving and assistance systems development for vehicle drivers has been subject of investigations from many years [1]. Usually, this kind of problem is decomposed into two different tasks: perception and control. We focus on the particular problem of the lateral control of a vehicle on its lane, or lane-keeping.

The perception task must provide an accurate and real-time estimation of the orientation and lateral position of the vehicle within its lane. Since the road is defined by white lane-markings, a camera is used as a perception tool. The control task requires computing, in real time, the wheel angle in such a way that the vehicle stays at the center of the lane.

A key problem is to decide about the choice of the parameters transmitted between the control and perception modules. This raises the question of designing an approach which integrates both control and perception aspects. A popular technique in control theory is the well-known Kalman filtering. Kalman theory is very powerful and convenient, but it is based on the assumption that the state and the measures are Gaussian random variables. Most of the time, outputs of vision processes are far from the Gaussian assumption. This has been shown in several vision problems, for instance [3][4][2]. This leads us to consider robust Kalman theory when measures are not Gaussian, but corrupted by *outliers*. Various algorithms [5][6][7] were proposed to tackle the problem of robust Kalman filtering. The first algorithm proposed in [6] is difficult to apply in practice. Alternatives described in [5] and [7] outline an approach leading to weighted least squares algorithms. However, these approaches are restricted to a small number of convex functions, while the one we propose here is valid for a large class of not necessarily convex functions. Also, contrary to our approach, the estimation step of the algorithm in [5][7] is not iterative.

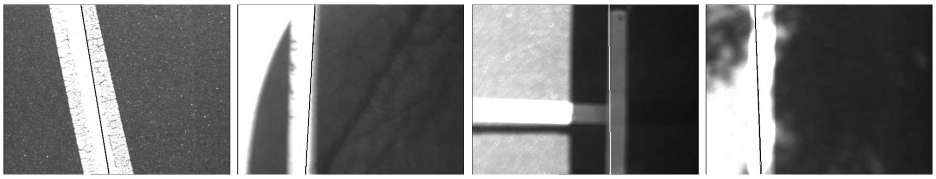
We propose here an overview of the problem based on Lagrange multipliers for deriving the equations of the robust Kalman filtering leading to a iterative reweighted least squares algorithm. To our knowledge, in the existing derivations, the explanation of why the robust Kalman filtering is not exact is rarely discussed. The main advantage of this derivation, which is equivalent to the half-quadratic approach [3][4], is to allow us to see two levels of approximations. One consists in assuming a Gaussian summary of the past and the other concerns the covariance matrix of the estimated state at every time step. Different possible approximate covariance matrices are proposed and experimentally compared.

The paper is organized as follows. First, we describe the system inboard the vehicle, and show that the features we are extracting from every image are not Gaussian. Second, for the sake of clarity, we gradually review least squares, recursive least squares, and Kalman filtering theory, and finally derive the robust Kalman filtering. Finally, we show the advantages of the designed robust Kalman filtering for the estimation of lane-markings position on perturbed road images and provide a comparison between the different approximate covariance matrices.

2 Image Feature Extraction

Fig. 1. Side camera system.

We have developed a system for measuring the lateral position and orientation of a vehicle using a vertical camera on its side. Due to the camera specifications and position, the accuracy of this system should be about 2 cm in position. Fig. 1 shows a first version of the system. A second version, where the camera is inside the left side mirror, is in progress. The image plane is parallel to the road surface, and the camera is mechanically aligned with the vehicle axis. This geometry reduces the calibration of the system to very simple manipulations.

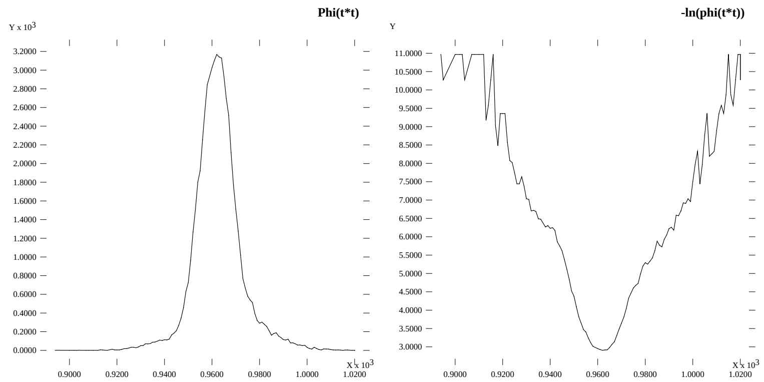


(a) (b) (c) (d)

Fig. 2. Typical image without perturbation (a), and perturbations due to another markings, shadows, lighting conditions (b) (c) (d). Solid lines are the fitted lane-markings centers assuming Gaussian noise.

Fig. 2(a) displays a typical example of images observed by the camera. The seen lane-marking is very close to a straight line, even in curves. Images (b), (c) and (d) are examples of perturbations due to other markings and lighting conditions. The image processing consists in first, extracting features in each newly grabbed image and second, in robustly fitting a line (or another kind of curve, as described in the next section). The first step is required for real time processing. The set of extracted features must provide a summary of the image content relevant to the application. On every line of an image, a lane-marking is approximatively seen as a white hat function on the intensity profile. Lane-marking centers, on every image line, are chosen as the extracted features.

Following the approach in [8], we want to reduce as much as possible the effect of low image contrast on the extracted features. Consequently, we have to design a detector which is relatively invariant to contrast changes. When the threshold on the intensity is reduced, features in images are numerous, and a criterion for selecting these becomes mandatory. We believe that selection based on geometrical considerations is a better alternative than selection based on intensity contrast. Since the system is calibrated, the feature extraction is performed on the width of lane-markings which is assumed to range between 8 and 23 cm.



(a) (b)

Fig. 3. *(a) Distribution of errors, (b) negative of its logarithm as a function of noise b.*

The obtained set of points is used by the line fitting. The question arises about the probability distribution function (pdf) of the extracted points around the true line. Most of the time, this pdf is assumed to be Gaussian. In Fig. 3(a), the measured pdf from a sequence of more than 100 real images is displayed. The pdf is not Gaussian, since Fig. 3(b) does not look like a parabola. Indeed, deeper investigations have shown that the curve in Fig. 3(b) can be very well approximated by $\phi(b^2) = \sqrt{1 + \frac{b^2}{\sigma^2}} - c$ with $\sigma = 5$, in a range of $[-20, 20]$ pixels around the minimum. For a good approximation on a larger range, a linear combination of the same kinds of functions with different values of σ seem to be needed.

3 Robust Estimation Framework

We consider that the lane-marking centers, extracted as described in the previous section, are noisy measurements of an underlying curve explicitly described as a function of one of its image coordinates:

$$y = \sum_{i=0}^{d} f_i(x) a_i = X(x)^t A \tag{1}$$

where (x, y) are the image coordinates of a point on the curve, $A = (a_i)_{0 \leq i \leq d}$ is the coefficient vector of the curve parameters, and $X(x) = (f_i(x))_{0 \leq i \leq d}$ is a vector of basis functions of the image coordinate x. In the context of our application, the basis functions are chosen as $f_i(x) = x^i$. The underlying curve is therefore a polynomial of degree d (i.e, a line when $d = 1$, a parabola when $d = 2$). Other bases may be used with their corresponding advantages or disadvantages.

In our model, the vertical coordinate is chosen as the x and assumed non-random. Thus only the other coordinate of the extracted point, y, is considered as a noisy measurement, i.e. $y = F(x)^t A + b$. In all that follows, the measurement noise b is assumed independent and identically distributed (iid), and centered.

For an intuitive understanding, we make a gradual presentation of the robust Kalman framework. Non-recursive least squares fitting is first recalled. Then, robust estimators are presented based on Lagrange multipliers approach and approximate inverse covariance matrices are proposed. In the fourth subsection, we introduce recursive and robust least squares (recursive least squares is a simple case of Kalman filter, using a constant state model). Finally, the robust Kalman filter is described.

3.1 Least Squares Fitting

First, we remember the very simple situation where only one image is observed and where the noise b is Gaussian. The goal is to estimate the curve parameters A_{LS} on the whole n extracted points (x_i, y_i), $i = 1, ..., n$. This issue is also known as a regression problem. Let $\mathcal{A}$ denote the underlying curve parameters we want to approximate with A_{LS}. 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, is:

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

For simpler equations, from now, we denote $X_i = X(x_i)$. We can write the probability of the whole set of points as the product of the individual probabilities:

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

where p is the so-called likelihood of the point data set, given curve parameter A. $\propto$ denotes the equality up to a factor. Maximizing likelihood p with respect to A is equivalent to minimizing the negative of its logarithm, namely:

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

It is the so-called least squares error. Since the fitting error is quadratic and positive, the minimization of e_{LS} is equivalent to canceling the vector of its first derivative with respect to A. It gives the well-known normal equations:

$$XX^t A = XY \tag{3}$$

where $Y = (y_i)_{1 \le i \le n}$ is the vector of y coordinates, the matrix $X = (X_i)_{1 \le i \le n}$ is the *design matrix*, and $S = XX^t$ is the *scatter matrix* which is always symmetric and positive. If S is definite, (3) has the unique solution $A_{LS} = S^{-1}XY$. Computing the best fit A_{LS} simply requires solving the linear system (3). As seen before, it is also the Maximum Likelihood Estimate (MLE).

Since only Y is random, the expectation of A_{LS} is $\overline{A_{LS}} = S^{-1}X\overline{Y}$. The point coordinates in $\overline{Y}$ correspond to points exactly on the underlying curve, thus $\mathcal{A} = S^{-1}X\overline{Y}$. Therefore, $\overline{A_{LS}}$ equals $\mathcal{A}$, i.e. the estimator A_{LS} of $\mathcal{A}$ is unbiased. The covariance matrix C_{LS} of A_{LS} is $\overline{(A_{LS} - \overline{A_{LS}})(A_{LS} - \overline{A_{LS}})^t} = S^{-1}X\overline{(Y - \overline{Y})(Y - \overline{Y})^t}X^tS^{-t}$. We have $\overline{(Y - \overline{Y})(Y - \overline{Y})^t} = \sigma^2 I_d$, since the noise b is iid with variance σ^2. I_d denotes the identity matrix of size $n \times n$. Finally, the inverse covariance matrix of A_{LS} is deduced:

$$C_{LS}^{-1} = \frac{1}{\sigma^2} S = Q_{LS} \tag{4}$$

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} with respect to A.

Finally, since e_{LS} is minimum in A_{LS} with second derivative matrix Q_{LS}, (2) can be rewritten as:

$$p \propto e^{-\frac{1}{2}(A - A_{LS})^t Q_{LS} (A - A_{LS})} \tag{5}$$

As clearly shown on Fig. 2, least squares fitting does not provide correctly fit curves in the presence of image perturbations.

3.2 Robust Fitting

We still assume that only one image is observed, and that measurement noises are iid and centered. But now, the noise is not assumed Gaussian, but having heavier tails. The heaviest observed noise 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:

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

Similarly to the half-quadratic approach [3][4], $\phi(t)$ is assumed:

- **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 the ones used in M-estimator approach for the convergence proof. Indeed in [9], the convergence proof requires that $\rho(b) = \phi(b^2)$ is convex. In our case, the concavity and monotony of $\phi(t)$ implies that $\phi'(t)$ is bounded, but $\phi(b^2)$ is not *necessarily convex* with respect to b. Note that, the pdf of Sec. 2, observed in practice on real data, verifies these three assumptions.

Following [9], the role of this ϕ function is to saturate the error in case of an important measurement noise $|b_i| = |X_i^t A - y_i|$, and thus to lower the importance of outliers. The scale parameter, σ, sets the distance from which a measurement noise has a good chance to be considered as outliers. Notice that with certain ϕ, the associated pdf cannot be integrated on its support. Without difficulties, a bounded support with fixed bounds can be introduced to maintain the statistical interpretation of the fitting.

Following the same MLE approach than for least squares, the problem is set as the minimization with respect to A of the robust error:

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

Notice that the Gaussian case corresponds to the particular case in which $\phi(t) = t$, but this last function does not strictly agree with assumption (H2). $e_{LS}(A)$ is indeed a limit case of $e_R(A)$. Contrary to the Gaussian case, the previous minimization is in general not quadratic. This last minimization can be done iteratively using the Gradient or Steepest Descent algorithms. But, since $\phi(b^2)$ and thus $e_R(A)$ are not necessarily convex, these algorithms can be relatively slow when the gradient slope is near zero. Indeed, the speed of convergence is only linear, when quasi-Newton algorithms achieve a quadratic speed of convergence. But generally, with quasi-Newton algorithms, the convergence to a local minimum is not sure. Therefore, we prove next that the used quasi-Newton algorithm always converges towards a local minimum. A global minimum can be obtained using simulated annealing, despite an expensive computational cost [3].

We now explain how this e_R can be solved iteratively, using the well known quasi-Newton algorithm named iterative reweighted least squares. The same algorithm is also a particular case obtained with the half-quadratic approach [4]. First, we rewrite $e_R(A)$ as the search for a saddle point of the associated Lagrange function. Then, the algorithm is obtained as a alternated minimization of the dual function.

First, we rewrite the minimization of $e_R(A)$ as the maximization of $-e_R$. This will allow us to later 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 $w_i = (\frac{X_i^t A - y_i}{\sigma})^2$. 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 precious since it allows us to introduce the Lagrange multipliers. Indeed using (H1), $-e_R(A)$ can be seen as the minimization with respect to $W = (w_i)_{1 \le i \le n}$ of:

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

subject to n constraints $h_i(A, W) = w_i - (\frac{X_i^t A - y_i}{\sigma})^2 \le 0$.

For any A, we now focus on the minimization of $E(A, W)$ with respect to W only subject to the n constraints $h_i(A, W) \le 0$, with respect to W only. This problem is well-posed because it is a minimization of a convex function subject to convex constraints. Therefore using the classical Kuhn and Tucker's theorem [10], if a solution exists, the minimization of $E(A, W)$ with respect to W is equivalent to the search of the unique saddle point of the Lagrange function of the problem:

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

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

$$-e_R(A) = \min_{w_i} \max_{\lambda_i} L_R(A, W, \lambda_i) \tag{6}$$

Notice that the Lagrange function L_R is now quadratic with respect to A, contrary to the original error e_R. Using the saddle point property, we can change the order of the variables w_i and λ_i in (6). $L_R(A, W, \lambda_i)$ being convex with respect to W, it is equivalent to search for a minimum of $L_R(A, W, \lambda_i)$ with respect to W and to have its first derivative zero. Thus, we deduce:

$$\lambda_i = \phi'(w_i) \tag{7}$$

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:

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

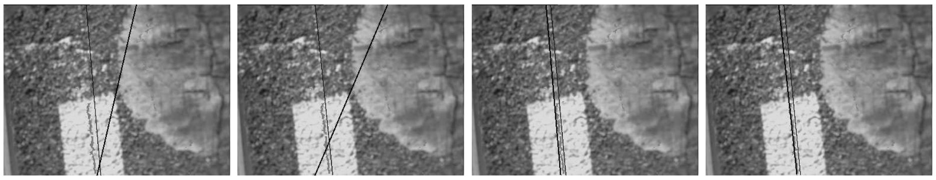
$\mathcal{E}(A, \lambda_i) = -L_R(A, {\phi'}^{-1}(\lambda_i), \lambda_i)$ is the dual function. The dual function is convex with respect to A. $\mathcal{E}$ is also convex with respect to λ_i (Indeed, $\frac{\partial^2 \mathcal{E}}{\partial \lambda_i^2} =$

$-\frac{1}{\phi''(\phi'^{-1}(\lambda_i))}$). Since $e_R(b^2)$ is not convex, it is not necessary that $\mathcal{E}$ is convex with respect to A and λ_i. Therefore, $\mathcal{E}(A, \lambda_i)$ does not have a unique minimum.

An alternate minimization of the dual function leads to the classical robust algorithm, used in the half-quadratic and M-estimator approaches:

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

The convergence test can be also performed on the error variation. A test on a maximum number of iterations can be added too. It can be shown that the previous 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 [11]. Using the previous Lagrange function, this proves that the previous algorithm is globally convergent, i.e, it converges towards a local minimum of $e_R(A)$ for all initial A_0s which are not a maximum of $e_R(A)$. As a quasi-Newton algorithm, it can be also proved that the speed of convergence of the algorithm around a local minimum is quadratic, when S is definite.



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

Fig. 4. Fitting on a real image assuming (a) Gauss, (b) quasi-Laplace, (c) Cauchy, and (d) Geman & McClure distributed noise. Thin black lines are the initial A_0's. Thick ones are the fitting results. See Sec. 4 for a definition of the pdfs.

Finally, Fig. 4 illustrates the importance of robust fitting in images with many outliers. The thin black lines depict the initial A_0's. The thicker ones are the fitting results A_R, assuming (a) Gauss, (b) quasi-Laplace, (c) Cauchy, and (d) Geman & McClure distributed noise. A correct fitting is achieved only with the last two pdfs which are not convex.

3.3 Covariance matrix in Robust Fitting

The covariance matrix C_R of the estimate A_R is required for a correct management of uncertainties in a recursive process. Contrary to the least squares case, where the covariance matrix was easy to compute using its definition, the estimation of C_R as the expectation of $(A_R - \overline{A_R})(A_R - \overline{A_R})^t$ is difficult in the robust framework, due to the non-linearities. An alternative is to use an approximation.

Similar to [9], p. 173-175, an approximation based on extending (4) is proposed. The inverse covariance matrix is approximated by the second derivative of e_R at the achieved minimum:

$$C_{R,Huber}^{-1} = \sum_{i=1}^{i=n} (2w_i\phi''(w_i) + \phi'(w_i))X_iX_i^t \tag{8}$$

where w_i is computed once the minimum of e_R is achieved. The value $\frac{d^2\phi((\frac{b}{\sigma})^2)}{db^2} = 2w\phi''(w) + \phi'(w)$ is not always positive, since $\phi((\frac{b}{\sigma})^2)$ is not necessarely convex with respect to b. Nevertheless, the second derivative of e_R with respect to A at A_R is a positive matrix since A_R achieves a minimum. This property is a necessary condition for the matrix being interpreted as a covariance matrix.

In [5][7], another approximation is implicitly used in the context of approximate robust Kalman filtering. The proposed approximate inverse covariance matrix can be seen as the second derivative of $-L_R$ with respect to A, at the achieved saddle point:

$$C_{R,Cipra}^{-1} = \sum_{i=1}^{i=n} \lambda_i X_i X_i^t \tag{9}$$

where λ_i is computed when the minimum of e_R is achieved. However, p. 175 of [9], Huber warns us against the use of this matrix (9).

Another approximation can be obtained if we forget that λ_i is a random variable. Let us rewrite the last equation of the robust algorithm as:

$$XRX^tA = XRY \tag{10}$$

where R is a $n \times n$ matrix with diagonal values λ_i, $1 \leq i \leq n$. Using these notations, the covariance matrix $C_{R,new1}$ is $\overline{(A_R - \overline{A_R})(A_R - \overline{A_R})^t}$ and equals $(XRX^t)^{-1}XR\overline{(Y-\overline{Y})(Y-\overline{Y})^t}R^tX^t(XRX^t)^{-t}$. Recalling from Sec. 3.1, that $\overline{(Y-\overline{Y})(Y-\overline{Y})^t} = \sigma^2 I_d$, we deduce:

$$C_{R,new1}^{-1} = \frac{1}{\sigma^2}(XRX^t)^t(XR^2X^t)^{-1}(XRX^t) \tag{11}$$

We also propose, without justification, another approximation:

$$C_{R,new2}^{-1} = \frac{1}{\sigma^2}\sum_{i=1}^{i=n} \lambda_i^2 X_i X_i^t \tag{12}$$

Now, the question is "what is the best choice for an approximate inverse covariance matrix?" A theoretical study is out of the scope of this paper, but we provide an experimental comparison in Sec. 4.

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 t, we have to fit its n_t data points $(x_{i,t}, y_{i,t})$, $i = 1, ..., n_t$. Of course, assuming that every point in every image is iid and centered, it is clear that we could directly apply what is explained in the two previous sections, on the whole data set. However, it is better to take advantage of the sequential arrival of images and deploy a recursive algorithm, in particular for saving memory space and number of computations, especially in the context of real time processing.

Recursive Least Square Fitting: When least squares error is used, 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:

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

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

$$e_{rLS,t}(A) = \frac{1}{2}(A - A_{rLS,t-1})^t Q_{rLS,t-1}(A - A_{rLS,t-1}) + \frac{1}{2\sigma^2} \sum_{i=1}^{i=n_t} (X_{i,t}^t A - y_{i,t})^2 \tag{13}$$

Using (5), 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 (5) its second order matrix is $Q_{rLS,t}$. Taking second derivative of (13), we deduce:

$$Q_{rLS,t} = Q_{rLS,t-1} + \frac{1}{\sigma^2} S_t \tag{14}$$

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 $e_{rT,t}$ with respect to A:

$$Q_{rLS,t} A_{rLS,t} = Q_{rLS,t-1} A_{rLS,t-1} + \frac{1}{\sigma^2} T_t \tag{15}$$

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 to 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 (14).
4. Compute the current fit $A_{rLS,t}$ by solving the linear system (15).
5. If a new dataset 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 can be crucial in real time applications, since the complexity for solving the linear system is $O(d^2)$, when it is $O(d^3)$ for a matrix inverse.

Note that (14) 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 (15) is unique. Indeed, $Q_{rLS,t}$ is definite for any t, even if S_t in (14) is not. This is equivalent to the *Ridge Regression* regularization [12]. 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.

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 rewrite (13) excepted for a very narrow class of function ϕ, that do not satisfy our assumptions (see sufficient statistics in [13]). Moreover, to obtain a solution without approximation, the computation of the weights λ_i would require storing all past observed points in memory up to the current time step. For real time application, 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 Sec. 3.3. It is usual to consider $A_{rR,t-1}$ as Gaussian with a covariance matrix $C_{rR,t-1} = Q^{-1}_{rR,t-1}$, while it is a 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 minimize by following the same approach than in Sec. 3.2. If Huber's (8) approximate is used, the new approximate inverse covariance matrix at time step t is:

$$Q_{rR,t,Huber} = \sum_{i=1}^{i=n_t} (2w_{i,t}\phi''(w_{i,t}) + \phi'(w_{i,t}))X_{i,t}X_{i,t}^t + Q_{rR,t-1,Huber} \quad (16)$$

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 using Cipra's (9) and our approximations (11) and (12).

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 ($1 \le i \le n_t$), compute the auxiliary variable $w_{i,j,t} = (\frac{X_{i,t}^t A_{j-1,t} - y_{i,t}}{\sigma})^2$,

4. Solve the linear system

$$(\sum_{i=1}^{i=n_t} \phi'(w_{i,j,t}) X_{i,t} X_{i,t}^t + Q_{rR,t-1}) A_{j,t} = \sum_{i=1}^{i=n_t} \phi'(w_i) X_i y_i + Q_{rR,t-1} A_{rR,t-1}$$

5. If $\|A_{j,t} - A_{j-1,t}\| > \epsilon$, 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 (16) or similar. If a new dataset 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, and a model of the link between input and output.

We can identify state A_t at time t with $A_{rLS,t}$ or $A_{rR,t}$, depending of the measurement noise pdf. As in Sec. 3.1, 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:

$$\begin{cases} A_t = U_t A_{t-1} + V_t + u \\ Y_t = \quad X_t A_t + v \end{cases} \tag{17}$$

When v is Gaussian, (17) models the classical Kalman (subscript K). When v is non Gaussian, (17) 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 (17), 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:

$$\hat{Q}_{K,t} = \Sigma^{-1} - \Sigma^{-1} U_t (U_t^t \Sigma^{-1} U_t + Q_{K,t-1})^{-1} U_t^t \Sigma^{-1} \tag{18}$$

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 (13), the prediction is used as a Gaussian prior on A. The associated error, to be compared with (13), 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 filtering 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 before with recursive least squares, generally, an exact solution of the robust Kalman is not achievable. The two levels of approximations must be performed. Like in Sec. 3.4, we assume that $A_{RK,t-1}$ is approximatively Gaussian, and its inverse covariance matrix is given by one of the approximations of Sec. 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, the Huber's approximation (16), translates as:

$$Q_{RK,t,Huber} = \sum_{i=1}^{i=n_t}(2w_{i,t}\phi''(w_{i,t}) + \phi'(w_{i,t}))X_{i,t}X_{i,t}^t + \hat{Q}_{RK,t} \qquad (19)$$

Other approximate inverse covariance matrix can be derived using Cipra's (9) and our approximations (11) and (12).

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 (18),
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 $w_{i,j,t} = (\frac{X_{i,t}^t A_{j-1,t} - y_{i,t}}{\sigma})^2$,
5. solve the linear system

$$(\sum_{i=1}^{i=n_t}\phi'(w_{i,j,t})X_{i,t}X_{i,t}^t + \hat{Q}_{RK,t})A_{j,t} = \sum_{i=1}^{i=n_t}\phi'(w_i)X_i y_i + \hat{Q}_{RK,t}\hat{A}_{RK,t}$$

6. If $\|A_{j,t} - A_{j-1,t}\| > \epsilon$, 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 (19) or similar. If a new dataset is available, increment t, and go to 2.

Note that in [5][7], 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 [7] are binary. This corresponds to a truncated Gaussian pdf, violating (H0). In such a case, the choice of the scale parameter becomes critical: a small variation of the scale parameter can produce a very different solution.

As a conclusion, the Lagrange multipliers approach (and half-quadratic approach) of robust fitting allows us to have new insight in why robust Kalman filtering provides approximate estimates. Robust Kalman is not exact because: the amount of past data cannot be reduced without loss of information, and the covariance matrix of the predicted state is an approximation. Contrary to [5][7], this formulation also suggests that it is important to iteratively search for the best solution A_t at every time steps.

4 Experiments

α	Name	pdf $\propto e^{-\frac{1}{2}\phi(b^2)}$	error=$\phi(b^2)$	weight=$\phi'(t)$
1	Gauss	$\propto e^{-\frac{1}{2}b^2}$	b^2	1
0.5	quasi-Laplace	$\propto e^{-\frac{1}{2}\sqrt{1+b^2}}$	$2(\sqrt{1+b^2}-1)$	$\frac{1}{\sqrt{1+t}}$
0	Cauchy	$\propto \frac{1}{1+b^2}$	$ln(1+b^2)$	$\frac{1}{1+t}$
-1	Geman & McClure [3]	$\propto e^{\frac{1}{2}\frac{1}{1+b^2}}$	$\frac{b^2}{1+b^2}$	$\frac{1}{(1+t)^2}$

Table 1. Correspondence between particular values of α and classical ϕs and pdfs proposed in the literature.

We have restrict ourselves in the choice of ϕ to the following one parameter family of functions:

$$\phi_\alpha(t) = \frac{1}{\alpha}((1+t)^\alpha - 1)$$

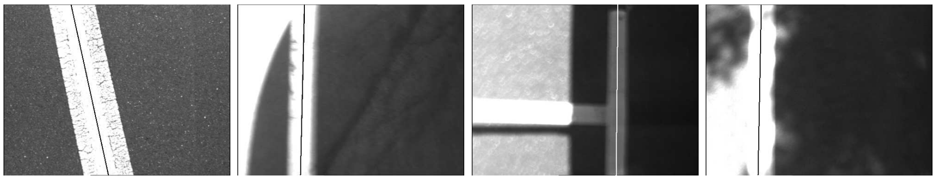
These functions verify the three assumptions (H0), (H1), and (H2), when $\alpha < 1$.

This family is very convenient, since it allows us to catch many of the classical ϕs and pdfs proposed in the literature. Tab. 1 illustrates this fact. Notice that the pdf obtained in the experiments of Sec. 2 corresponds to $\alpha = 0.5$. The pdf obtained for $\alpha = 0.5$, also known as the hypersurface function, is a good differentiable approximation of Laplace's pdf. Thus we have preferred to name it the quasi-Laplace function.

Name	$\sqrt{C_{00}}$	$\sqrt{C_{11}}$	$\sqrt{C_{00}}$ rel. Std.	$\sqrt{C_{11}}$ rel. Std.
C_{new1}	0.138	0.00474	7.8%	10.6%
C_{Cipra}	0.162	0.00555	9.8%	13.2%
C_{Huber}	0.189	0.00647	10.5%	14.1%
C_{new2}	0.190	0.00653	13.1%	17.6%
reference	0.195	0.00688		

Table 2. Comparison between the covariance matrices obtained with various approximations. The relative standard deviations are also shown.

A simulation was performed using 50000 fits on simulated 101 noisy points along a line with true parameters $a_0 = 100$ and $a_1 = 1$. a_0 is the pose of the line and a_1 is its slope. The noise pdf for each sample is Cauchy $\propto \frac{1}{1+(\frac{b}{\sigma})^2}$ with $\sigma = 1$. The Cauchy noise was simulated by applying the function $tan(\frac{\pi}{2}v)$ on v, a uniform noise on $[-1, 1]$. The variance of the Cauchy pdf is not defined, thus the simulated noise can have very large values (outliers). Robust fits were obtained using the Cauchy-distributed pdf of Sec. 3.2. For every fit, the Huber's, Cipra's, and ours approximate covariance matrices were computed and averaged. These are denoted C_{Huber}, C_{Cipra}, C_{new1} and C_{new2}, respectively. The square roots of the averaged diagonal matrix components are shown in Tab. 2. The covariance matrix of the 50000 fits is also estimated and is the reference displayed in the last line of Tab. 2 (Monte-Carlo estimates).



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

Fig. 5. Typical image without perturbation (a), and perturbations due to another markings, shadows, lighting conditions (b) (c) (d). Solid lines are the fitted lane-markings centers assuming Geman & McClure noise.

Tab. 2 shows that the closest approximation is C_{new2}. All these approximations can be ordered in terms of proximity, with respect to the reference one, in the following order: C_{new2}, C_{Huber}, C_{Cipra} and C_{new1}. Notice that the closer to the reference one the matrix is, the larger its relative variation from one fit to another is. For instance, this variation is 7.8% for C_{new1} when it is 13.1% for C_{new2}. Clearly, the choice of the approximation is a trade-off between accuracy and stability. We also notice that all the approximations under-estimate the true matrix. A different weighting of the two right terms in (16) and (19) can be introduced for correcting this.

Finally, we show in Fig. 5, the same images than in Fig. 2 using Geman & McClure noise assumption. Unlike the Gaussian assumption, the obtained fits are correct even in presence of important perturbations due to other lane-markings and difficult lighting conditions.

5 Conclusion

In this paper, we have reviewed the problem of making Kalman filtering robust to outliers, in a unified framework. The link with Lagrange multipliers yields a revised half-quadratic theory and justifies the use of iterative reweighted least squares algorithms in M-estimator theory even for non-convex $\rho(b) = \phi(b^2)$. Moreover, in contrast to previous works, we do not restrict ourselves to a single potential function but the half-quadratic framework is valid for a large class of functions, involving non-convex *and hence more robust* ones. We have shown that, as soon as non-Gaussian likelihoods are involved, two levels of approximation are needed. First, in contrast with the non-robust case, there is no obvious closed-form expressions for the covariance matrix. After reviewing classical solutions, we proposed new approximations and experimentally studied their behavior in terms of accuracy and stability. An accurate covariance matrix is very important to tackle the problem of missing data on a long sequence of images, an important subject for future investigations. Second, to design a recursive filter in the robust case, 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 least squares algorithm to converge. Further exploration than presented in [5] is needed to treat the case where the noise involved in the state-transition equation is non-Gaussian. Here the challenge is to derive an integrated and consistent framework.

References

1. 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 S.161-16*, volume 727, 1986.
2. P.H.S. Torr and A. Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.
3. D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Trans. on Pattern Analysis and Intelligence*, 14(3):367–383, 1992.
4. P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *Trans. on Image Processing*, 6(2):298–311, 1997.
5. T. Cipra and R. Romera. Robust kalman filtering and its application in time series analysis. *Kybernetika*, 27(6):481–494, 1991.
6. C. Masreliez. Approximate non-gaussian filtering with linear state and observation relations. *IEEE Transactions on Automatic Control*, 18:107–110, February 1975.
7. R. Rao. Robust kalman filters for prediction, recognition, and learning. Technical Report TR645, University of Rochester, 1996.
8. F. Guichard and J.-P. Tarel. Curve extraction combining perceptual grouping and a kalman like fitting. In *IEEE International Conference on Computer Vision*, Kerkyra, Greece, 1999.
9. P. J. Huber. *Robust Statistics.* John Wiley and Sons, New York, New York, 1981.
10. M. Minoux. *Mathematical Programming: Theory and Algorithms.* Chichester: John Wiley and Sons, 1986.
11. D. G. Luenberger. *Introduction to linear and nonlinear programming.* Addison Wesley, 1973.
12. 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.
13. H. W. Sorenson. *Parameter Estimation.* Marcel Dekker, Inc., New York, New York, 1980.