

# Evaluation of Robust Fitting Based Detection

Sio-Song Ieng<sup>1</sup>, Jean-Philippe Tarel<sup>2</sup>, and Pierre Charbonnier<sup>3</sup>

<sup>1</sup> LIVIC (LCPC-INRETS), 14, route de la minière, Bât 824,  
F-78000 Versailles-Satory, France  
ieng@inrets.fr

<sup>2</sup> ESE (LCPC), 58, boulevard Lefèbvre,  
F-75732 Paris Cedex 15 France  
tarel@lcpc.fr

<sup>3</sup> LRPC de Strasbourg,  
11, Rue Jean Mentelin, BP 9,  
F-67200 Strasbourg, France  
Pierre.Charbonnier@equipement.gouv.fr

**Abstract.** Low-level image processing algorithms generally provide noisy features that are far from being Gaussian. Medium-level tasks such as object detection must therefore be robust to *outliers*. This can be achieved by means of the well-known M-estimators. However, higher-level systems do not only need robust detection, but also a confidence value associated to the detection. When the detection is cast into the fitting framework, the inverse of the covariance matrix of the fit provides a valuable *confidence matrix*.

Since there is no closed-form expression of the covariance matrix in the robust case, one must resort to some approximation. Unfortunately, the experimental evaluation reported in this paper on real data shows that, among the different approximations proposed in literature that can be efficiently computed, none provides reliable results. This leads us to study the robustness of the covariance matrix of the fit with respect to noise model parameters. We introduce a *new non-asymptotic approximate covariance matrix* that experimentally outperforms the existing ones in terms of reliability.

## 1 Introduction

In modern applications, such as intelligent transportation systems, cameras and their associated detection algorithms are increasingly considered as specialized sensors. As for every sensor, the vision measure must be accompanied with a *confidence* value in order to be integrated into a higher level system. This is especially important when safety aspects are involved. But then the point is: how can this evaluation be performed?

Many image analysis algorithms, such as image segmentation or curve detection, can be at least partially formalized as *fitting* problems. Least-squares fitting, the well known technique based on the assumption of Gaussian noise, is widely used, and it provides the exact covariance matrix of the obtained fit. The

inverse of this matrix can be used as a *confidence matrix*. However, it is common knowledge that, *in real applications, the noise is seldom Gaussian*, making the fitting task more difficult. Alternatively, using the robust framework, it is possible to deal with heavy tailed noise models, but this leads to non-linear equations and iterative algorithms such as reweighted least-squares, in the context of *M-estimators* [1].

Although robust fitting algorithms have been extensively investigated, the problem of deriving a confidence matrix has seldomly been considered in image analysis. Indeed, due to non-linearities, an exact derivation of the covariance matrix is believed to be intractable in the robust framework and approximations are required. One way to tackle the problem is to study the asymptotic behavior of robust estimators: Huber [1] proposes several such approximate covariance matrices. The evaluation of these matrices was only performed on synthetic data. In a different context, namely robust Kalman filtering, where the question of the predicted covariance matrix becomes of major importance, other approximations were proposed in [2] and in [3], but without justification. Again, the evaluation of these matrices was performed only on synthetic data.

We started our study by an experimental comparison of the approximate covariance matrices already proposed in the literature. The results of this comparison on real and simulated data showed that none of them gives sufficiently reliable results when noise is far from being Gaussian, which led us to derive a new approximate covariance matrix. Unlike those proposed by Huber, it is not an asymptotic covariance matrix. However, its experimental performances are much more satisfactory.

Robust fitting using M-estimators is summarized in Section 2, where the choice of the noise model and its parameters is also discussed. Then in Section 3, a new approximation is derived and the experimental comparison of approximate covariance matrices is described.

## 2 Fitting Based Detection

We focus, as an application, on lane-markings detection in images [3]. For each image the detection algorithm consists of a feature extraction step followed by a robust curve fitting. The feature extractor scans each row  $x$  of the image and provides a series of coordinates  $(x_i, y_i)$ ,  $i = 1, \dots, n$  that should correspond to the center of each lane-marking. The detection can then be seen as a fitting problem, using explicit curves along one axis of the image. The assumed link between a couple of coordinates  $(x_i, y_i)$  is

$$y_i = X_i^t A + b_i. \quad (1)$$

In this model, the measurement noise,  $b$  is assumed to be centered, independent and identically distributed (iid). The vector  $X_i = (x_i^k)$ ,  $k = 0, \dots, d$  is the vector of monomials. While a straight line is sufficient as a lane-marking model for near-field cameras, other basis functions could be used to model more complex

lane-marking curves, provided the relationship remains linear with respect to the curve parameters,  $A = (a_k), k = 0, \dots, d$ .

In the remainder of the section, the robust fitting procedure is reviewed and its performances and limitations are discussed.

### 2.1 Feature Noise Model

An experimentally convenient way of modeling observed noise distribution is to use the so-called Smooth Exponential Family  $F_{\alpha,s}(b)$  of functions introduced in [3]:

$$F_{\alpha,s}(b) \propto \frac{1}{s} e^{-\frac{1}{2} \phi_{\alpha}(\frac{b^2}{s^2})} \tag{2}$$

where

$$\phi_{\alpha}(\frac{b^2}{s^2}) = \frac{1}{\alpha} ((1 + \frac{b^2}{s^2})^{\alpha} - 1).$$

The two parameters of this family are  $s$  and the power  $\alpha$ . The former is the scale parameter, while the latter specifies the shape of the noise distribution. Indeed,  $\alpha$  allows a continuous transition between well-known statistical laws such as Gauss ( $\alpha = 1$ ), and Geman & McClure [4] ( $\alpha = -1$ ). Another advantage of this family is that it only contains differentiable functions allowing robust fitting, as proved in [3]. Moreover,  $F_{\alpha,s}$  can always be normalized on a bounded support, so it can still be seen as a probability distribution function (pdf). In this family, when  $\alpha$  decreases, the probability of observing large errors (outliers) increases.

### 2.2 Robust Fitting

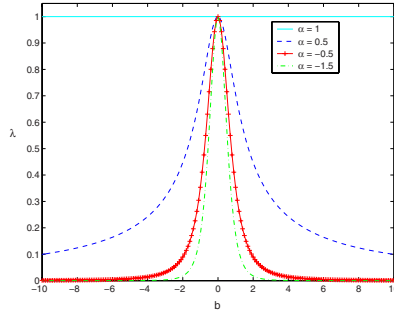
Following the MLE approach and assuming the noise model (2), the problem is equivalent to minimizing, with respect to  $A$ , the error:

$$e(A) = \frac{1}{2} \sum_{i=1}^{i=n} \phi_{\alpha}((\frac{X_i^t A - y_i}{s})^2).$$

A local minimum is achieved by the classical alternate minimisation scheme:

1. Initialize  $A_0$ , and set  $j = 1$ .
2. For all indexes  $i$  ( $1 \leq i \leq n$ ), compute weights  $\lambda_{i,j} = \phi'_{\alpha}((\frac{X_i^t A_{j-1} - y_i}{s})^2)$ .
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  $\|A_j - A_{j-1}\| > \epsilon$ , increment  $j$ , and go to 2, else a local minimum is achieved.

This algorithm was first introduced for M-estimators by Huber [1]. A proof of the local convergence of the previous algorithm, using Kuhn and Tucker's theorem is described in [3]. When  $\alpha = 0$ , it is easy to show that generalized T-Student [5] pdfs are used as noise model.



**Fig. 1.** Variations of weight  $\lambda$  with respect to the value  $b$  of the noise, for different values of  $\alpha$ . Outliers effect is strongly reduced by this weight during the fitting.

### 2.3 Influence of $\alpha$ and $s$ on Fits

The choices of the parameters of the noise model,  $\alpha$  and  $s$ , are now discussed.

Fig. 1 shows that the weight  $\lambda$  becomes less sharply peaked when  $\alpha$  is decreasing. The same kind of figure is obtained with increasing  $s$ . As a consequence, the lower  $\alpha$  or  $s$  is, the lower the effect of outliers on the result is.

But one must be careful not decrease  $\alpha$  or  $s$  too much. Indeed, with too small an  $\alpha$  or  $s$ , the number of local minima increases and the curve fitting algorithm has a higher chance of being trapped in 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 [6]. So, how can the scale and power parameters be correctly estimated?

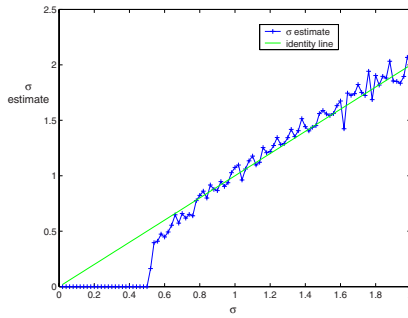
When the pdf support is unbounded ( $\alpha \geq 0$ ) and other noise parameters are known, by deriving the negative log-likelihood with respect to  $s$ , we get the following implicit equation in  $\hat{s}$ :

$$\hat{s}^2 = \frac{1}{n} \sum_{i=1}^{i=n} \phi' \left( \left( \frac{X_i^t A - y_i}{\hat{s}} \right)^2 \right) (X_i^t A - y_i)^2. \quad (3)$$

This equation means that the MLE estimate  $\hat{s}$  of  $s$  is a fixed point. When  $\alpha < 0$ , a similar expression, accounting for bounded support pdf's can be also derived. Other scale estimators such as MAD [7] have been proposed.

The MLE estimator of the scale must be used with care. Surprisingly, when data has little noise, we observed that the MLE estimate of  $s$  is clearly underestimated. This is a general problem, due to the finite precision of feature detectors (in our case, it provides discrete image positions). In Fig. 2, we simulate this effect by generating a Cauchy noise ( $\propto \frac{1}{1+(\frac{b}{s})^2}$  or  $\alpha = 0$ ) with a fixed scale  $s$ , and rounding it. Then, the MLE scale  $\hat{s}$  is estimated from this data, assuming a Cauchy noise with unknown scale. Fig. 2 confirms our observations on real data: when the true scale  $s$  is lower than one pixel,  $\hat{s}$  is clearly under-estimated.

This suggests that during its estimation, the scale must not be allowed to take small values (i.e. lower than one pixel). This also implies that it is better



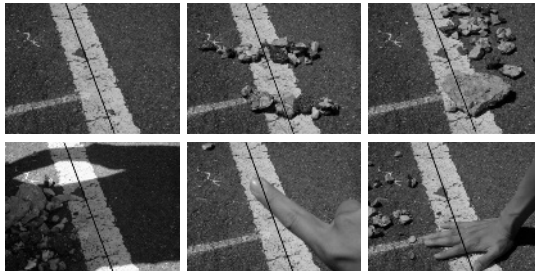
**Fig. 2.** The estimated scale  $\hat{s}$  versus the true scale  $s$  for Cauchy noise. We can notice that  $\hat{s}$  is under-estimated by the MLE estimator when  $s < 1$ , due to rounding the data.

not to estimate the scale between each step of the robust fitting, contrary to what was suggested in [1]. Indeed, an under-estimated scale can freeze the fit in its current state, while it has not yet converged.

The MLE approach, when applied to the estimation of  $\alpha$ , assuming  $s$  is known, does not seem to lead to closed-form estimators. Nevertheless, this estimation may always be performed by minimizing numerically the likelihood with respect to  $\alpha$ , using a minimizer such as a Gradient Descent algorithm. In our experiments, we have obtained  $\hat{\alpha} = 0.05$  on the residual noise collected on 150 images after fitting.

### 2.4 Robustness with Respect to $\alpha$ and $s$

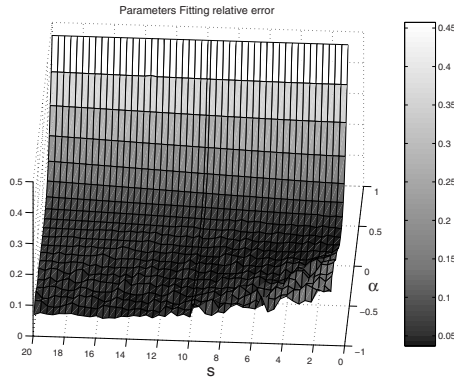
By nature, robust fitting algorithms are not very sensitive to outliers. However, another question of importance is their robustness to an inappropriate choice of the noise model parameters. In order to evaluate the effect of this kind of modeling error on the fitting, we run the following experiment:



**Fig. 3.** Six images from the set of 150 images used in the experiments. The black straight line is the reference fit.

1. Collect a set of 150 images showing the same lane-marking in the same position but with different perturbation (Several images are shown in Fig. 3).

2. The reference lane-marking is accurately measured by hand. In this experiment fits are straight lines.
3. For each image of the set, fits are estimated for pairs of  $(\alpha, s)$  ranging on a grid. The initial curve parameters,  $A_0$ , are set to random values.
4. The relative error between the fits and the reference is averaged on the image set for each pair  $(\alpha, s)$ . Fig. 4 shows the obtained error surface.



**Fig. 4.** Errors between the fit and the reference for different  $\alpha$  and  $s$ . Notice the very broad ranges of  $\alpha$  and  $s$  that lead to a flat valley.

Observe how the valley along an  $\alpha$  cross section is very large compared to a  $s$  cross section. In general, it appears that  $\alpha$  and  $s$  can be chosen in a large range. Moreover, when  $\alpha$  is set in the range of correct values,  $s$  can vary within a very large range without really modifying the resulting fits. This robustness to the choice of parameters probably explains why robust fitting can be used with success in many practical applications.

### 3 Evaluation of the Detection

It is common knowledge that in practical applications, an estimate must be accompanied with some measure of *confidence* in order to be correctly used by higher level systems (e.g. Kalman filters in tracking applications [3]). In the statistical estimation framework, this confidence measure is naturally given by (the inverse of) the covariance matrix. This is why we have formulated detection as a fitting problem in the previous section. With Gaussian noise, computing the covariance matrix is straightforward. In the robust framework, however, a correct covariance matrix estimate is more difficult to obtain. There are two main reasons for that. First, there is no known closed-form solution, but only approximations. In this section, we focus on approximations that can efficiently be computed from the data in hand during the fitting. Second, it turns out that an accurate choice of the noise model parameters is much more critical for the covariance matrix estimation than for the fitting, as demonstrated below.

### 3.1 Confidence Matrices

A complete review of the many different approaches for approximating covariance matrices is out of the scope of this paper. 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 we will evaluate.

The simplest approximate covariance matrix  $C_{Cipra}$ , named Cipra’s covariance matrix in [3], is:

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

where  $\lambda_i$  are the weights when robust fitting converged. Another fast-to-compute approximation is proposed in [3]:

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

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

In chapter 7 of [1], Huber derives an asymptotic covariance matrix and proposes three other approximate covariance matrices:

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

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

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

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:

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

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 sequel.

In our experiments with real data, the previously proposed covariance matrices did not provide reliable estimations of the detected lane-markings. This is

experimentally shown in the next section. We propose the following new approximate covariance matrix which is justified in the Appendix:

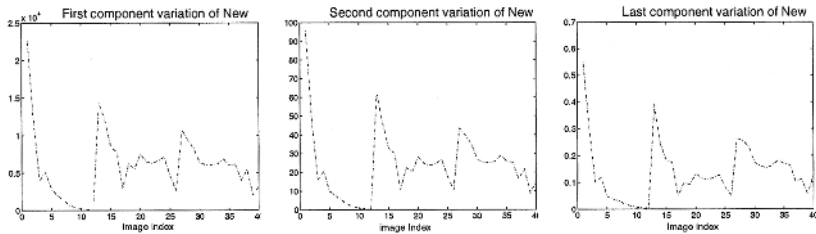
$$C_{New} = \frac{\sum_{i=1}^{i=n} \lambda_i b_i^2}{\sum_{i=1}^{i=n} \lambda_i - Trace(O_2 O_1^{-1})} O_1^{-1} O_2 O_1^{-1}. \tag{4}$$

This approximation is not an asymptotic covariance matrix, unlike Huber’s approximations. The advantage is thus that  $C_{New}$  does not require a high number  $n$  of data points to be applied. It relies on assuming non random weights  $\lambda_i$ . If the weights were considered as random variables, deriving the covariance matrix would become too involved.

### 3.2 Experimental Covariance Matrix Comparison

The six previous covariance matrix estimates are compared on average on 150 real images (partially shown in Fig. 3). The experimental comparison process is the same as in Sec. 2.4, 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 averaged on the image set for each  $(\alpha, s)$ .



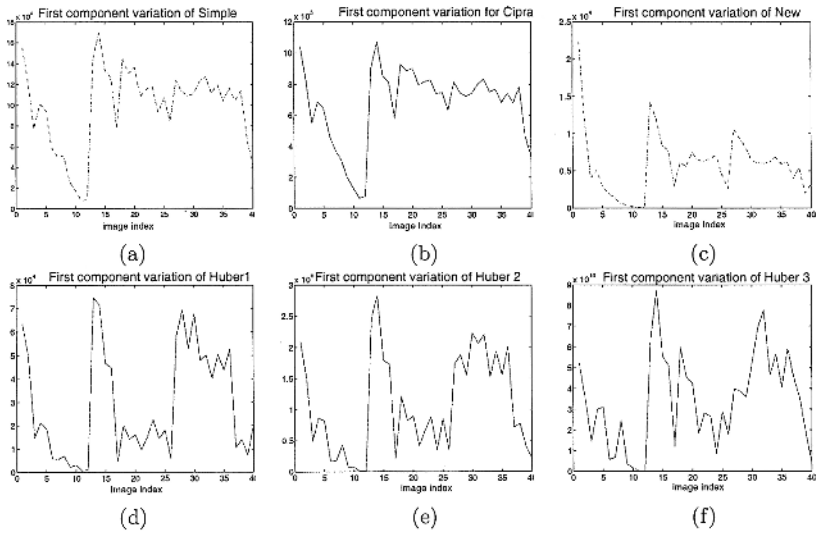
**Fig. 5.** These curves represent the variation of the three components of the  $2 \times 2$  symmetric matrix  $C_{New}$  versus the index of the image. The first and the last figures are associated to the diagonal components.

Fig. 5 illustrates the fact that the three components of a same matrix vary in the same way. Thus, without loss of generality, the analysis can be performed on only the first diagonal component.

Fig. 6 shows the value of the first component of the six confidence approximations for  $\alpha = 0.2$  and  $s = 2$ . These variations are relatively similar, but the orders of magnitude can vary greatly from one approximation to another.

In Tab. 1, the relative errors with respect to the reference matrix for the different approximate covariance matrices are compared with  $(\alpha, s) = (0.2, 2)$ .





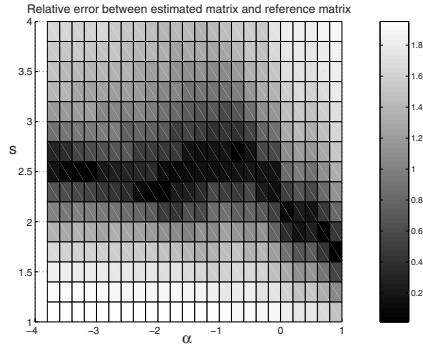
**Fig. 6.** These curves represent the first component of the different confidence matrices:  $C_{Simple}$  (a),  $C_{Cipra}$  (b),  $C_{New}$  (c),  $C_{Huber1}$  (d),  $C_{Huber2}$  (e) and  $C_{Huber3}$  (f). We notice that the values are of different orders of magnitude, but they vary in a similar way. This estimation is performed for  $\alpha = 0.2$  and  $s = 2$ .

Similar results are obtained for other  $(\alpha, s)$  in the valley of Fig. 4 where fits are accurate. The relative error is calculated with  $\frac{2|C_{ij} - C_{ref,ij}|}{C_{ij} + C_{ref,ij}}$ . Thus, 200% corresponds to the worst case where the value of  $C_{ij}$  is far away from the reference value  $C_{ref,ij}$ . This table shows that only  $C_{New}$  achieves a usable estimation of the confidence matrix. This is also illustrated by Fig. 6 where  $\alpha = 0.2$  and  $s = 2$  and where only  $C_{New}$  has the correct order of magnitude. As a consequence,  $C_{New}$  appears to be more robust than the other approximations.

In Fig. 7, the relative error between the first component of the confidence matrix  $C_{New}$  and the reference is displayed for different values of  $\alpha$  and  $s$ . For the other components, similar relative error maps are obtained. This relative error map shows that a relative error of 20% in the vicinity of  $s = 2.5$  for  $\alpha < 0$

**Table 1.** Relative errors in percentage on components  $C_{11}$ ,  $C_{22}$  and  $C_{12}$  of the confidence matrix with respect to the reference, for the six covariance matrix estimates, for  $(\alpha, s) = (0.2, 2)$ .

Type	$C_{11}$ rel.err.	$C_{22}$ rel.err.	$C_{12}$ rel.err.
Cipra	195%	198%	196%
Huber1	156%	183%	192%
Huber2	200%	200%	200%
Huber3	200%	200%	200%
Simple	98%	99%	101%
New	64%	41%	43%



**Fig. 7.** Relative error of  $C_{New}$  versus  $(\alpha, s)$  with respect to reference covariance. With the new approximate covariance matrix, a relative error of 20% is achieved contrary to the others.

can be achieved. This was not obtained for the other approximations. This is due to the fact that  $C_{New}$  is more robust to the choice of  $\alpha$ . Nevertheless, in contrast to parameter fitting, a correct estimate of  $s$  remains important for a correct  $C_{New}$ .

In Sec. 2.3, the average power parameter  $\alpha$  was estimated on the residual noise collected from 150 images as  $\hat{\alpha} = 0.05$ . Since the noise scale  $s$  is not the same from one image to another, it is better to estimate the average scale as the average of the scales estimated on each image with  $\alpha$  fixed to 0.05. The average model noise parameters obtained are  $(\hat{\alpha}, \hat{s}) = (0.05, 2.02)$ , which is in the valley observed in Fig. 7. Therefore, our experiments are consistent.

Similar synthetic experiments were performed for higher degree curves. Higher degree requires to introduce fitting regularisation. 10000 perturbed data sets with Cauchy noise were generated and fitted by a 2nd degree curve (3 parameters) assuming the true noise model. We obtained an average error of  $-5\%$  for  $C_{new}$ ,  $+9\%$  for  $C_{simple}$ ,  $-16\%$  for  $C_{cipra}$ . Huber approximates are not of the same order of magnitude. This extends to higher degree results obtained on real data.

## 4 Conclusion

In this paper, we considered object detection as a statistical estimation problem. Then, the inverse covariance matrix naturally provides a measure of the confidence that can be associated to the detection result, and the non-Gaussian nature of the noise distributions that can be encountered in practice can be dealt with using robust fitting.

In this context, the important question which we have addressed is the *choice of the noise distribution model parameters*. We have experimentally shown that estimated fits remain relatively reliable even with a slightly incorrect noise model. On the contrary, correct covariance matrix estimates generally need fine tuning of the noise model parameters, which is difficult to achieve in practice.

We thus derived a new approximate covariance matrix (4) which is more robust to incorrect noise model parameters. This new approximate covariance matrix can be used to great advantage in many applications, including detection.

**Acknowledgement.** This work is supported by INRETS and région de l'île de France. The authors are grateful to LIVIC technical team for its helps in the experiments.

## References

1. P. J. Huber. *Robust Statistics*. John Wiley and Sons, New York, New York, 1981.
2. T. Cipra and R. Romera. Robust kalman filtering and its application in time series analysis. *Kybernetika*, 27(6):481–494, 1991.
3. 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, Denmark, 2002.
4. S. Geman and D. McClure. Statistical methods for tomographic image reconstruction. *Bull. Int. Stat. Inst.*, LII(4):5–21, 1987.
5. 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.
6. R. Dahyot. *Appearance based road scene video analysis for management of the road network*. PhD thesis, University of Strasbourg I, 2001.
7. F.R. Hampel, P.J. Rousseeuw, E.M. Ronchetti, and W.A. Stahel. *Robust Statistics*. John Wiley and Sons, New York, New York, 1986.

## Appendix: $C_{New}$ Covariance Matrix Derivation

Let us justify  $C_{New}$ . The covariance matrix of the estimated parameter vector  $\hat{A}$  is defined as the expectation  $E[(\hat{A} - A)(\hat{A} - A)^t]$ , where  $A$  is the true parameter vector. We rewrite step 3 of the robust algorithm as  $\hat{O}_1 \hat{A} = \hat{B}$  with  $\hat{B} = \sum_{i=1}^{i=n} \hat{\lambda}_i \hat{y}_i X_i$ . For the true parameters, we have the similar equation  $O_1 A = B$  where  $B = \sum_{i=1}^{i=n} \lambda_i y_i X_i$ . In the following, we remove the constraint that  $\hat{\lambda}_i$  is a random variable and thus  $\hat{\lambda}_i = \lambda_i$ . This has three main consequences. First, it implies  $\hat{O}_1 = O_1$ . Second, since  $b_i$  is assumed centered, from  $\hat{y}_i = X_i^t A + b_i$  we deduce  $y_i = X_i^t A$ , and thus  $\hat{B} - B = \sum_{i=1}^{i=n} \lambda_i b_i X_i$ . Since the  $b_i$ 's are independent random variables, this last equation implies  $E[(\hat{B} - B)(\hat{B} - B)^t] = \sum_{i=1}^{i=n} \lambda_i^2 E[b_i^2] X_i X_i^t = E[b^2] O_2$ . Third, using the previous equations, the covariance matrix can be approximated by  $O_1^{-1} E[(B - \hat{B})(B - \hat{B})^t] O_1^{-t}$  or

$$C_{New} = E[b^2] O_1^{-1} O_2 O_1^{-t}, \tag{5}$$

after substitution of the covariance matrix of  $\hat{B}$ . The point 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 leads to bad estimates when the noise model is not correctly chosen. A better idea, leading to estimates more robust to errors in the choice of the noise model, is to derive  $E[b^2]$  from the data residuals  $b_i$  using (3).

We can now derive the equation between  $E[b^2]$  and  $E[\hat{s}^2]$ . (3) is rewritten as  $\hat{s}^2 = \frac{1}{n} \sum_{i=1}^{i=n} \lambda_i (\hat{y}_i - X_i^t \hat{A})^2$ . By introducing the true parameter vector  $A$ , inside the sum, we get  $\hat{s}^2 = \frac{1}{n} \sum_{i=1}^{i=n} \lambda_i (b_i + X_i^t (A - \hat{A}))^2$ . To calculate  $E[\hat{s}^2]$  as a function of  $E[b^2]$ , we expand the squared term in the previous equation. Using (5), we deduce

$$E[\hat{s}^2] = \frac{1}{n} \sum_{i=1}^{i=n} (E[b^2] \lambda_i + E[b^2] \lambda_i X_i^t O_1^{-1} O_2 O_1^{-1} X_i + 2 \lambda_i X_i^t E[(A - \hat{A}) b_i]). \quad (6)$$

Since  $b_i$  is centered, thus  $E[(A - \hat{A}) b_i] = -E[\hat{A} b_i]$  in (6). The estimated parameter vector obtained by fitting is  $\hat{A} = O_1^{-1} \sum_{k=1}^{k=n} \lambda_k X_k X_k^t \hat{y}_k = O_1^{-1} \sum_{i=1}^{i=n} \lambda_k X_k (X_k^t A + u_k)$ . Therefore  $E[\hat{A} b_i]$  can be expanded as  $E[\hat{A} b_i] = O_1^{-1} \sum_{k=1}^{k=n} \lambda_k X_k E[u_k b_i]$ . The independence of  $b_i$  allows us to simplify the last equation in  $E[\hat{A} b_i] = O_1^{-1} \lambda_i X_i E[b^2]$ . As a consequence (6) is now:

$$E[\hat{s}^2] = \frac{E[b^2]}{n} \left( \sum_{i=1}^{i=n} \lambda_i + \sum_{i=1}^{i=n} \lambda_i X_i^t O_1^{-1} O_2 O_1^{-1} X_i - 2 \sum_{i=1}^{i=n} \lambda_i^2 X_i^t O_1^{-1} X_i \right). \quad (7)$$

Using the property  $Trace(AB) = Trace(BA)$ , we can rewrite the second and third terms in (7) as  $\sum_{i=1}^{i=n} \lambda_i X_i^t O_1^{-1} O_2 O_1^{-1} X_i = \sum_{i=1}^{i=n} \lambda_i^2 X_i^t O_1^{-1} X_i = Trace(O_2 O_1^{-1})$ . This implies the simple result  $E[\hat{s}^2] = \frac{E[b^2]}{n} \left( \sum_{i=1}^{i=n} \lambda_i - Trace(O_2 O_1^{-1}) \right)$ . Using (3) as an approximate value of  $E[\hat{s}^2]$ , we deduce the following estimate of  $E[b^2]$ :

$$E[b^2] = \frac{\sum_{i=1}^{i=n} \lambda_i b_i^2}{\sum_{i=1}^{i=n} \lambda_i - Trace(O_2 O_1^{-1})}. \quad (8)$$

The substitution of (8) in (5), results in the proposed approximate covariance matrix (4).