Certainties in low-level Operations

Hagen Spies
Computer Vision Laboratory
Dept. of Electrical Engineering, Linköping University
581 83 Linköping, Sweden
*h*spies@isy.liu.se

Abstract

A method to compute certainties for low level operations such as averaging and derivation is presented. Based on a local signal model we identify convolution with a weighted least squares fit of this model to the data. The residual of the fit is then converted via its probability density function into a certainty (probability) that the observed signal can actually be described by the chosen model. The concept is illustrated on a few examples and it is shown how the obtained certainties may be used in further processing steps.

Keywords: signal models, certainty measure

1 Introduction

It has often been pointed out that in order for an algorithm to be useful in practice it should also produce some information about the quality of the computed results [1, 3]. While this is usually done for more complex operations such as orientation and velocity estimation such a procedure is not often applied on the very fist processing level, such as smoothing and derivative estimation. In this paper we present a method to assign certainties to such estimates based on how well local signal models describe the data. This is quite general as any linear filter can be identified with a least squares fit of some model.

Examples for the use of local signal approximations are the facet model [4] and polynomial expansion [2]. In contrary to these approaches we are not primarily interested in a good signal approximation here but rather aim to assign a statistically sound certainty measure to standard low level operations. Another difference is that we focus on very local, typically 3×3, approximations. For larger areas we integrate such local measurements in a second step utilizing the associated certainties.

Paper organization: Section 2 reviews the connection between least squares and convolution. In Sect. 3 it is shown how the residual can be estimated and converted into a probabilistic certainty measure (Sect. 3.2). Section 4 gives some results and illustrates how such information might be used in further processing steps.

2 Least Squares and Convolution

We begin by analysing the relationship between least squares estimation and convolution and show that they are essentially identical. A least squares problem is often defined as:

$$\min_{\boldsymbol{p}} \|\boldsymbol{A}\boldsymbol{p} - \boldsymbol{s}\|^2 \,, \tag{1}$$

where $\boldsymbol{A}$ is a given model matrix, $\boldsymbol{s}$ is a vector of noisy observations and $\boldsymbol{p}$ is the sought parameter vector. If $\boldsymbol{s}$ has uniform independent Gaussian noise the solution is given by:

$$\boldsymbol{p} = \boldsymbol{A}^+ \boldsymbol{s} \,, \tag{2}$$

where $\boldsymbol{A}^+$ denotes the pseudo-inverse. This general formulation also holds in the case of a rank deficient model matrix, in this case the minimum norm solution will be computed from the infinite number of solutions. However, in the following we will concentrate on the case where $\boldsymbol{A}$ has full column rank. Then the pseudo-inverse becomes:

$$\boldsymbol{A}^+ = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T \,. \tag{3}$$

In any case we obtain the parameter vector by multiplying the pseudo-inverse to the data vector. In terms of signal processing this is nothing but a correlation of the signal with the kernels $\boldsymbol{a}_i$, which are the rows of $\boldsymbol{A}^+$. Convolution kernels are readily obtained as the mirrored kernels.

Consider a simple 1D example where we approximate the signal by its mean and derivative value:

$$\tilde{\boldsymbol{s}} = s_m + s_x \boldsymbol{x} \,. \tag{4}$$

In a three point neighbourhood this can be written as:

$$\tilde{\boldsymbol{s}} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} s_m - s_x \\ s_m \\ s_m + s_x \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & -1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} s_m \\ s_x \end{bmatrix}}_{\boldsymbol{p}} \,. \tag{5}$$

Forming the pseudo-inverse we obtain:

$$(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ -\frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} \,. \tag{6}$$

Hence the mean is computed by a simple Box filter and the derivative by a central difference filter.

2.1 Weighted Least Squares

Clearly it is often advisable to give more influence to some points than to others. Either because of their distance to the evaluation point or because some points are more reliable than others. The more general normalized convolution method actually uses the product of both [9]. In order to do so we will introduce a weight on each equation given by a diagonal weight matrix $\boldsymbol{W} = \text{diag}(w_i)$. The minimization is performed with respect to the weighted least squares norm:

$$\min_{\boldsymbol{p}} \|\boldsymbol{s} - \boldsymbol{A}\boldsymbol{p}\|^2_{\boldsymbol{W}} = (\boldsymbol{s} - \boldsymbol{A}\boldsymbol{p})^T \boldsymbol{W} (\boldsymbol{s} - \boldsymbol{A}\boldsymbol{p}) \; . \quad (7)$$

The solution then becomes:

$$\boldsymbol{p} = (\boldsymbol{A}^T \boldsymbol{W} \boldsymbol{A})^+ \boldsymbol{A}^T \boldsymbol{W} \boldsymbol{s} \; . \quad (8)$$

Note that usually we can use the inverse instead of the pseudo-inverse here. However, even if $\boldsymbol{A}$ has full column rank the weights might be such that $(\boldsymbol{A}^T \boldsymbol{W} \boldsymbol{A})$ does not have full rank anymore.

If we use Binomial weights $w_1 = 0.25; w_2 = 0.5; w_3 = 0.25$ in the above 1D example we obtain:

$$(\boldsymbol{A}^T \boldsymbol{W} \boldsymbol{A})^{-1} \boldsymbol{A}^T \boldsymbol{W} = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ -\frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} \; . \quad (9)$$

The derivative filter remains unchanged but the mean is now computed using a Binomial instead of a Box filter. Somewhat more interesting is the 2D case, a linear model corresponds to a plane fit:

$$\tilde{\boldsymbol{s}}(x, y) = s_m + s_x x + s_y y \; . \quad (10)$$

In order to apply the formalism we form the signal vector of the considered neighbourhood by stacking the columns after each other. We will use Binomial weights:

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \rightarrow \boldsymbol{w} = \frac{1}{16} [1\,2\,1\,2\,4\,2\,1\,2\,1]^T \; , \quad (11)$$

and $\boldsymbol{W} = \text{diag}(\boldsymbol{w})$. The filters contained in the rows of $(\boldsymbol{A}^T \boldsymbol{W} \boldsymbol{A})^{-1} \boldsymbol{A}^T \boldsymbol{W}$ are the Binomial average and the Sobel filter in x and y direction (rearranged in 2D):

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \; ; \quad \frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$
$$\text{and} \quad \frac{1}{8} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \; . \quad (12)$$

Similarly does a quadratic model

$$\tilde{\boldsymbol{s}}(x, y) = s_m + s_x x + s_y y + \frac{1}{2} [x\; y] \begin{bmatrix} s_{xx} & s_{xy} \\ s_{xy} & s_{yy} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} , \quad (13)$$

lead to additional second order derivative filters. In general we obtain a derivative of Gaussian filter for a polynomial signal model and Gaussian weights.

Note that even for other filters we can compute a corresponding model $\boldsymbol{A}$ and weights $\boldsymbol{W}$. However this decomposition is usually not unique.

3 Residuals

As seen before the application of a linear filter can be regarded as a least squares fit of some model. Hence it is of great interest how well the fitted model describes the data. In other words we want to establish a certainty in the output as well as the parameter values.

If the signal model is given by $\boldsymbol{A}\boldsymbol{p} = \tilde{\boldsymbol{s}}$ then the model parameters are obtained from (8) which we rewrite in terms of the dual basis:

$$\boldsymbol{p} = \tilde{\boldsymbol{B}}^T \boldsymbol{s} \quad \text{with} \quad \tilde{\boldsymbol{B}} = \boldsymbol{W} \boldsymbol{A} (\boldsymbol{A}^T \boldsymbol{W} \boldsymbol{A})^+ \; . \quad (14)$$

The deviation of the signal from the model $\tilde{\boldsymbol{s}}$ in each point can be captured in an error vector:

$$\boldsymbol{e} = \tilde{\boldsymbol{s}} - \boldsymbol{s} = \boldsymbol{A} \tilde{\boldsymbol{B}}^T \boldsymbol{s} - \boldsymbol{s} = \underbrace{(\boldsymbol{A} \tilde{\boldsymbol{B}}^T - \boldsymbol{I})}_{\boldsymbol{R}} \boldsymbol{s} \; . \quad (15)$$

The elements of $\boldsymbol{e}$ can thus be computed by a correlation with the rows of $\boldsymbol{R}$. From the elements of $\boldsymbol{e}$ we can also compute a local error histogram. However this is not exploited further in this contribution.

Probably the most important quantity is the residual of the fit. If we are only interested in this mean squared error, we can directly compute it in the applied norm:

$$\begin{aligned} r &= |\boldsymbol{e}|^2_{\boldsymbol{W}} = \boldsymbol{e}^T \boldsymbol{W} \boldsymbol{e} = (\boldsymbol{R}\boldsymbol{s})^T \boldsymbol{W} (\boldsymbol{R}\boldsymbol{s}) \\ &= \boldsymbol{s}^T \boldsymbol{R}^T \boldsymbol{W} \boldsymbol{R}\, \boldsymbol{s} = \boldsymbol{s}^T \boldsymbol{Q} \boldsymbol{s} \; , \end{aligned} \quad (16)$$

which is a quadratic form defined by $\boldsymbol{Q}$.

3.1 Implementation

When implementing the above ideas it is advantageous to analyse the filter structure in more detail. We focus on the computation of the residual here.

From a generalization of the formula for the variance: $\sigma^2 = \bar{x^2} - \bar{x}^2$ we can compute the residual with only very little overhead without having to use the quadratic form $\boldsymbol{Q}$ explicitly. We rewrite r as:

$$\begin{aligned} r &= |\boldsymbol{s} - \tilde{\boldsymbol{s}}|^2_{\boldsymbol{W}} = (\boldsymbol{s} - \tilde{\boldsymbol{s}})^T \boldsymbol{W} (\boldsymbol{s} - \tilde{\boldsymbol{s}}) \\ &= \boldsymbol{s}^T \boldsymbol{W} \boldsymbol{s} + \tilde{\boldsymbol{s}}^T \boldsymbol{W} \tilde{\boldsymbol{s}} - 2 \boldsymbol{s}^T \boldsymbol{W} \tilde{\boldsymbol{s}} \; . \end{aligned} \quad (17)$$

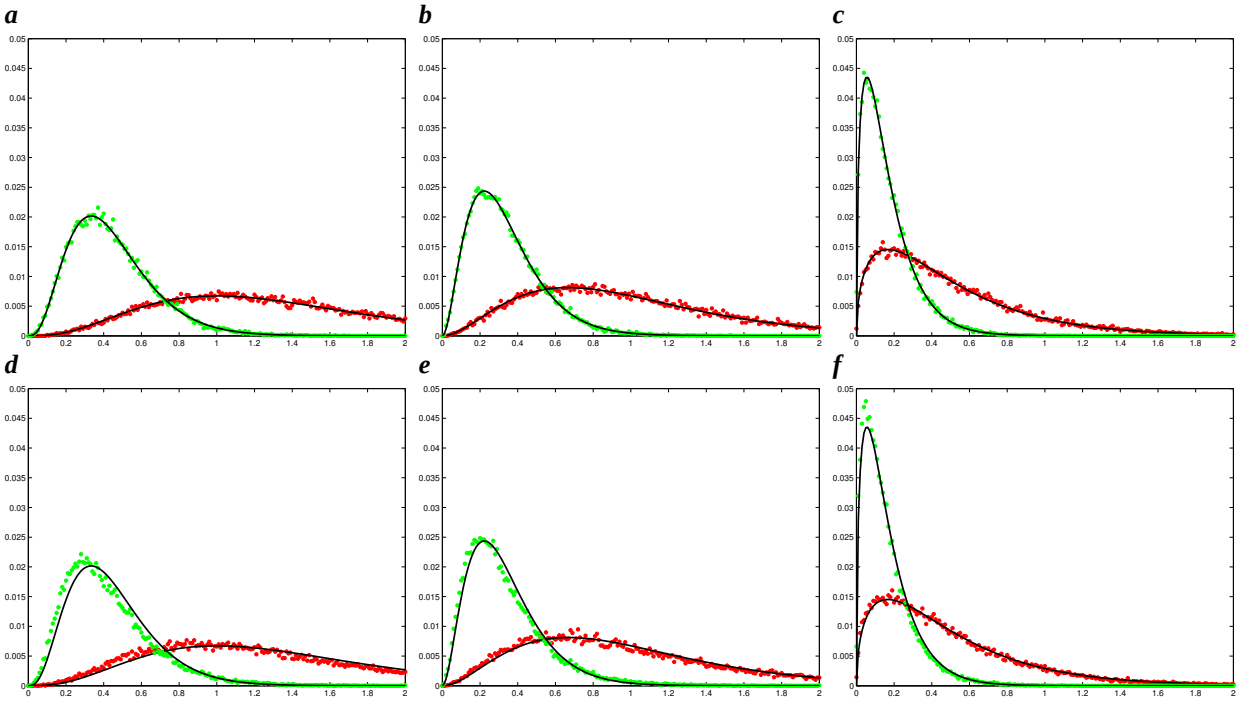

Figure 1: *Residual distributions on a constant signal, the two curves correspond to $\sigma^2 = 0.5$ and $\sigma^2 = 1.5$. Using uniform weights for different signal models in a 3×3 neighbourhood:* ***a*** *constant,* ***b*** *linear and* ***c*** *quadratic. Using Binomial weights:* ***d*** *constant,* ***e*** *linear and* ***f*** *quadratic model.*

The third term can be rewritten using Eq. (8) $\boldsymbol{s}^T\boldsymbol{W}\boldsymbol{A} = (\boldsymbol{A}^T\boldsymbol{W}\boldsymbol{s})^T = [(\boldsymbol{A}^T\boldsymbol{W}\boldsymbol{A})\,\boldsymbol{p}]^T = \boldsymbol{p}^T(\boldsymbol{A}^T\boldsymbol{W}\boldsymbol{A})$ and thus becomes:

$$\boldsymbol{s}^T\boldsymbol{W}\tilde{\boldsymbol{s}} = \boldsymbol{s}^T\boldsymbol{W}\boldsymbol{A}\boldsymbol{p} = \boldsymbol{p}^T(\boldsymbol{A}^T\boldsymbol{W}\boldsymbol{A})\boldsymbol{p} = \tilde{\boldsymbol{s}}^T\boldsymbol{W}\tilde{\boldsymbol{s}}\;. \tag{18}$$

Inserting $\tilde{\boldsymbol{s}} = \boldsymbol{A}\boldsymbol{p}$ also in the second term of (17) we finally obtain:

$$r = \boldsymbol{s}^T\boldsymbol{W}\boldsymbol{s} - \boldsymbol{p}^T(\boldsymbol{A}^T\boldsymbol{W}\boldsymbol{A})\,\boldsymbol{p}\;. \tag{19}$$

As $\boldsymbol{W}$ is diagonal the first term is the weighted average of the squared signal $\boldsymbol{s}^T\boldsymbol{W}\boldsymbol{s} = \sum_i \omega_i s_i^2$. The second term can easily be evaluated as $\boldsymbol{A}^T\boldsymbol{W}\boldsymbol{A}$ can be computed beforehand. At least for constant weights.

3.2 Certainty

Let's assume the noise in the signal to be zero mean Gaussian with variance σ^2. In computer vision applications it is common to assume a constant noise variance, however it is sometimes necessary to allow the noise to be spatially varying. This should either be taken into account in the weight matrix or by a preprocessing step.

If the model describes the data exactly the deviation in each point is caused solely by the signal noise. If we fit a model with m parameters in a neighbourhood of n pixels the sum of the normalized squared errors:

$$x = \sum_{i=i}^{n} \frac{\boldsymbol{e}_i^2}{\sigma_i^2} = \frac{1}{\sigma^2}\sum_{i=i}^{n} \boldsymbol{e}_i^2, \tag{20}$$

follows a χ^2-distribution with $n-m$ degrees of freedom. We now consider the case of equal weights, i.e. a Box filter, $\omega_i = \frac{1}{n}$. Then the residual $r = \frac{\sigma^2}{n}x$ follows a scaled χ^2-distribution [8] ($\mathrm{d}x = \frac{n}{\sigma^2}\mathrm{d}r$):

$$p(r) = \frac{n}{\sigma^2}\,\chi^2_{n-m}(\frac{n}{\sigma^2}\cdot r)\;. \tag{21}$$

This distribution together with the experimentally obtained residual distribution is shown for a few examples in Fig. 1. While we have have perfect agreement for uniform weights as expected, there is a slight deviation otherwise that depends on the non-uniformity of the weights. In the later experiments we concentrate on the 3×3 Binomial filter. In this case the agreement is sufficiently good. Hence we won't consider the more general case of arbitrary weights here.

When the model does not describe the data the observed residual distribution will not be given by (21). An example of a constant model applied to a linearly varying signal is shown in Fig. 2a. Using the linear model on the same data does again yield good agreement between theoretical and experimental distribution, see Fig. 2b.

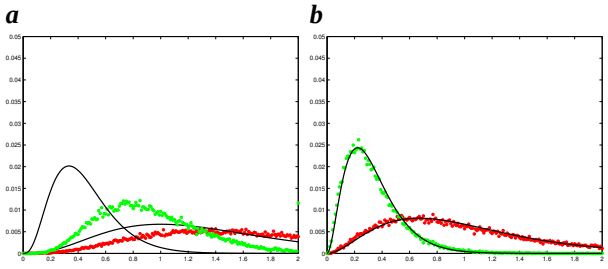

Figure 2: *Residual distributions on a linear signal, evaluated in a* 3×3 *neighbourhood, for* $\sigma^2 = 0.5$ *and* $\sigma^2 = 1.5$*. Using Binomial weights and a constant* ***a*** *, and linear model* ***b*** *.*

The expectation value of r is given by:

$$E(r) = \int_0^\infty r\, p(r)\mathrm{d}r = \int_0^\infty \frac{\sigma^2}{n}\, x\, \chi^2_{n-m}(x)\, \mathrm{d}x = \sigma^2 \frac{n-m}{n}, \quad (22)$$

where we used $\mathrm{d}r = \frac{\sigma^2}{n}\mathrm{d}x$ and that the expectation of the χ^2-distribution is given by the degrees of freedom ($E(x) = n - m$). For the variance we obtain, see Appendix A for details:

$$V(r) = \int_0^\infty [r - E(r)]^2\, p(r)\mathrm{d}r = 2(n-m)\frac{\sigma^4}{n^2}. \quad (23)$$

The probability that an observed residual value a is compatible with the model hypothesis is given by:

$$p(a) = \int_a^\infty p(r)\mathrm{d}r = 1 - \int_0^a p(r)\mathrm{d}r\ . \quad (24)$$

Inserting $p(r)$ from (21) yields:

$$p(a) = 1 - \int_0^a \frac{n}{\sigma^2}\, \chi^2_{n-m}(\frac{n}{\sigma^2}\cdot r)\mathrm{d}r = 1 - \int_0^{\frac{n}{\sigma^2}a} \chi^2_{n-m}(x)\mathrm{d}x\ , \quad (25)$$

which is readily computed using the well tabulated cumulative χ^2-distribution. This probability can directly be used as a certainty measure for the applied model. The thus obtained certainty as a function of the residual is shown in Fig. 3.

4 Experiments

We show the computed certainty for some test images in Fig. 4. For these images the actual noise level is not known and we choose $\sigma^2 = 1$ which is reasonable for standard CCD cameras [6].

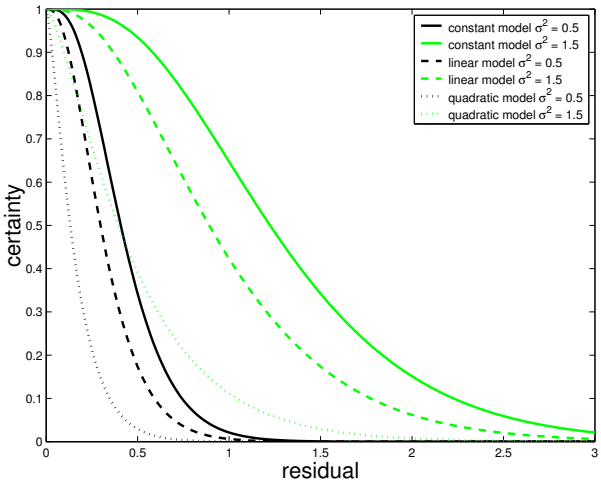

Figure 3: *Certainty as a function of the residual for different models and noise levels on a* 3×3 *neighbourhood.*

As one would expect, a higher order model can capture more variation and in general results in higher certainties. Increasing the neighbourhood size leads to larger low certainty areas near edges. However it also seems to distinguish better between uniform regions. The certainty is clearly lower at edges and in strongly textured regions.

We have shown how to compute the residual of fundamental low level estimates such as the mean, first and second derivatives and will now discuss some possible use of this information. In general it is of course very valuable to have information about the certainty of a given estimate. It can either be used directly for image enhancement, segmentation or edge/corner detection (Sect. 4.2). Additionally certainty information can be very useful for any further processing that combines the low level estimates into a higher order feature. Typical examples are the estimation of local orientation and optical flow.

4.1 Orientation Estimation

We denote the image derivatives by $\boldsymbol{s}_x$ and $\boldsymbol{s}_y$. The sample covariance matrix of the gradient, also called structure tensor, is given by:

$$\boldsymbol{T} = \begin{bmatrix} < \boldsymbol{s}_x^2 > & < \boldsymbol{s}_x \cdot \boldsymbol{s}_y > \\ < \boldsymbol{s}_x \cdot \boldsymbol{s}_y > & < \boldsymbol{s}_y^2 > \end{bmatrix} \quad (26)$$

where $< \cdot >$ denotes a local average, here computed by convolution with a Gaussian kernel $\boldsymbol{G}$: $< \boldsymbol{x} >= \boldsymbol{G} * \boldsymbol{x}$. Local orientation can be obtained from the eigenvector to the larger eigenvalue. In 2D the corresponding angle, in double angle representation, is given by [7]:

$$\tan(2\phi) = \frac{2 < \boldsymbol{s}_x \cdot \boldsymbol{s}_y >}{< \boldsymbol{s}_y^2 > - < \boldsymbol{s}_x^2 >}\ . \quad (27)$$

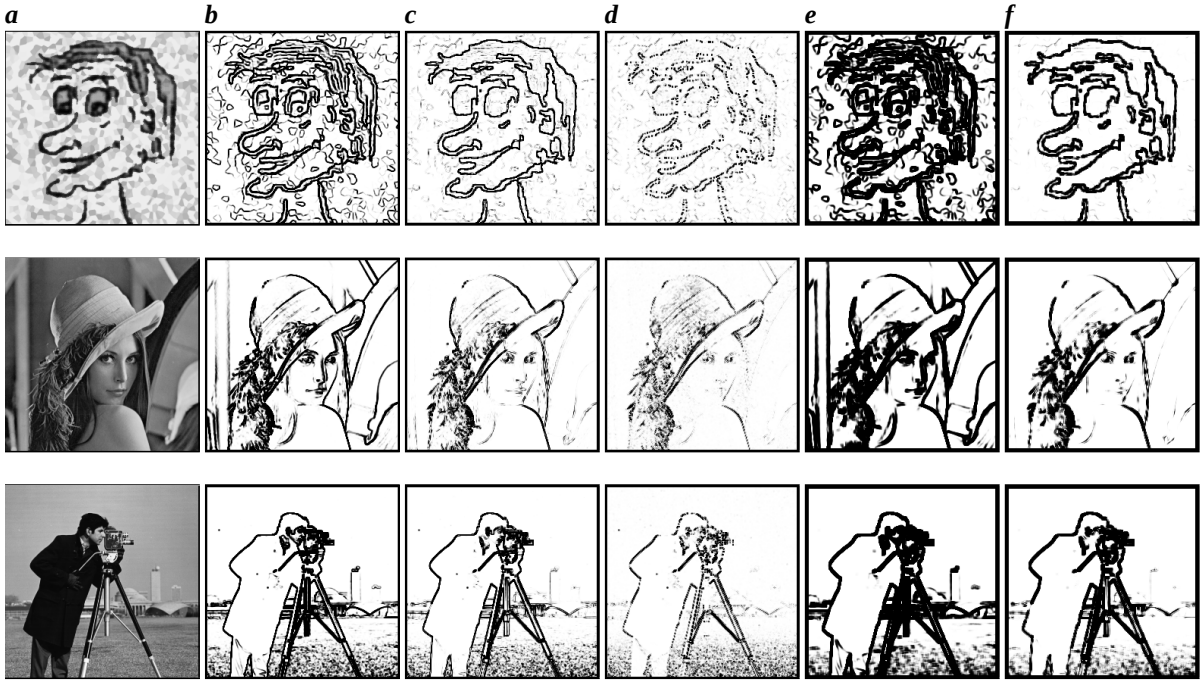

Figure 4: *Example certainties. Column* ***a*** *shows the used images and columns* ***b*** *-****d*** *the certainty for a constant, linear and quadratic model applied in a* 3×3 *neighbourhood using Binomial weights. Columns* ***e*** *and* ***f*** *show the certainty using* 5×5 *Gaussian (*$\sigma = 3$*) weights for a constant and quadratic model respectively.*

The certainty can easily be incorporated by replacing the average in (26) with a normalized average [3]:

$$< \boldsymbol{x} >_n = \frac{\boldsymbol{G} * (\boldsymbol{c} \cdot \boldsymbol{x})}{\boldsymbol{G} * \boldsymbol{c}} \; . \tag{28}$$

Figure 5 shows two examples with and without incorporating the certainty. There clearly is an improvement at image discontinuities, in the first example the discontinuity even vanishes completely. However in order not to average derivatives from both sides of the discontinuity the integration kernel can only be as large as the model neighbourhood. In the shown results we used a 3 tap Gaussian with $\sigma = 1$. For larger integration areas we will need to incorporate the presented ideas into a robust estimation scheme.

4.2 Corner and Edge Detection

We already saw in the examples given in Fig. 4 that the certainty nicely drops near edges and corners. In a similar fashion does the widely used Harris corner measure [5] employ a residual measure. This and the related measure given by Noble [10] essentially detect points where the determinant of the structure tensor (26) is high. For simple, i.e. only varying in one direction, signals this determinant is zero, hence we observe that they essentially measure the residual of a signal model that is constant in one direction.

However in order to detect edges and corners it is favourable to use the residual rather than the certainty. Due to statistical fluctuations the latter can sometimes even be small without the the presence of an edge. The residual is shown in Fig. 6 for a range of $[0 \; 4\sigma^2]$. Values higher than $4\sigma^2$ correspond to a clear model violation. Constant and linear model violations correspond to step edges, which can apparently be captured by a quadratic model. The failure of the quadratic model on the other hand corresponds to corners. Yet from Fig. 6d we observe that the distinction between corners and noise is not as clear as it is for the edges.

5 Conclusion

We showed how certainties can be attributed to common low level image processing operations. It has been illustrated that such information can be used to detect discontinuities, namely edges and corners, and thus for image segmentation. We also showed that the incorporation into further processing steps such as orientation estimation can make these estimates more robust in the presence of discontinuities.

Future work includes quantitative investigations to what respect the use of such low-level certainties can improve known algorithms for image enhancement, orientation and velocity estimation, edge and corner detection.

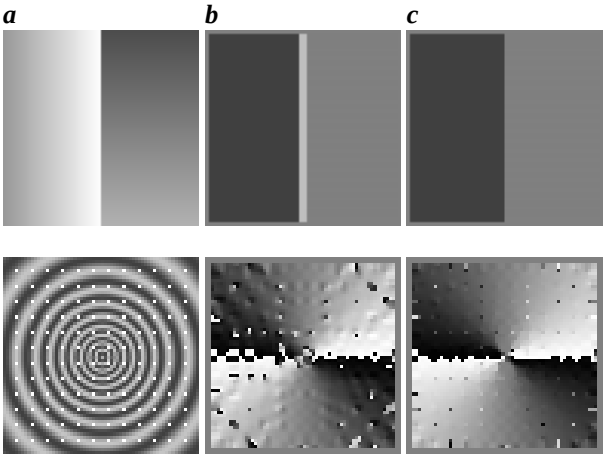

Figure 5: *Orientation estimation:* ***a*** *test images,* ***b*** *orientation angle without using certainty and* ***c*** *incorporating the certainty.*

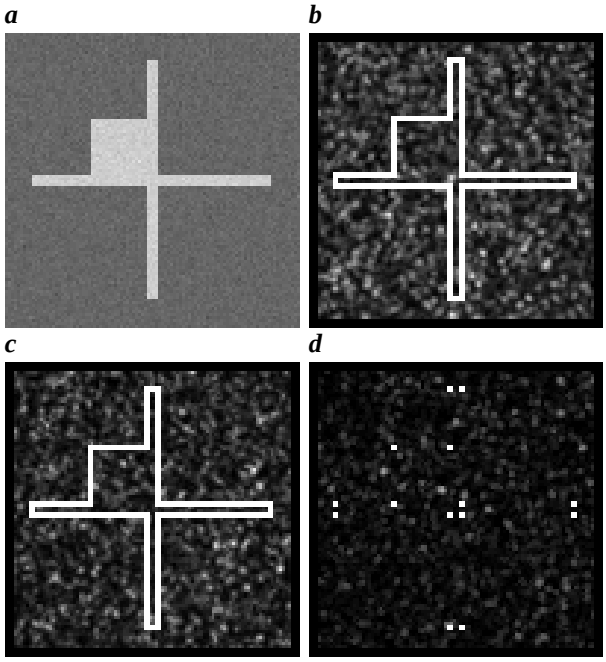

Figure 6: *Edge and corner detection:* ***a*** *test image and residual ($[0\ 4\sigma^2]$) for a constant* ***b*** *, linear* ***c*** *and quadratic model* ***d*** *.*

A Residual Variance

The variance of the residual $r = \sigma^2 \frac{x}{n}$ is defined as:

$$\begin{aligned} V(r) &= \int_0^\infty (r - E(r))^2\, p(r) \mathrm{d}r \\ &= \int_0^\infty (r - \frac{n-m}{n}\sigma^2)^2\, \frac{n}{\sigma^2}\, \chi^2_{n-m}(\frac{n}{\sigma^2} r)\, \mathrm{d}r \end{aligned}$$

Changing the variable to $x = \frac{n}{\sigma^2} r$ yields:

$$\begin{aligned} V(r) &= \int_0^\infty (\frac{\sigma^2}{n} x - \frac{n-m}{n}\sigma^2)^2\, \chi^2_{n-m}(x)\, \mathrm{d}x \\ &= \frac{\sigma^4}{n^2} E(x^2) - 2\frac{\sigma^4}{n^2}(n-m)E(x) + \frac{\sigma^4}{n^2}(n-m)^2, \end{aligned}$$

where we used $\int \chi^2 = 1$ in the last term. The expectation of x is $E(x) = n - m$ and that of x^2 can be determined from $V(x) = E(x^2) - (E(x))^2$. With $V(x) = 2(n-m)$ we get $E(x^2) = (n-m)^2 + 2(n-m)$. Hence we obtain:

$$\begin{aligned} V(r) &= \frac{\sigma^4}{n^2}[(n-m)^2 + 2(n-m) - 2(n-m)^2 + (n-m)^2] \\ &= 2(n-m)\frac{\sigma^4}{n^2}\,. \end{aligned}$$

Acknowledgements: This work has been funded by a fellowship within the Postdoc-Programme of the German Academic Exchange Service (DAAD).

References

[1] J. L. Barron, D. J. Fleet, and S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[2] G. Farnebäck. Very high accuracy velocity estimation using orientation tensors, parametric motion, and simultaneous segmentation of the motion field. In *ICCV*, pages 171–177, Vancouver, July 2001.

[3] G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic, 1995.

[4] R. M. Haralick and L. Watson. A facet model for image data. *Computer Graphics and Image Processing*, 15(2):113–129, February 1981.

[5] C. G. Harris and M. J. Stephens. A combined corner and edge detector. In *Fourth Alvey Vision Conference*, pages 147–151, Manchester, UK, 1988.

[6] B. Jähne. *Practical Handbook on Image Processing for Scientific Applications*. CRC Press, 1996.

[7] B. Jähne. *Digital Image Processing*. Springer, Heidelberg, Germany, 5 edition, 2002.

[8] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Praxis*. Elsevier Science, Amsterdam, 1996.

[9] H. Knutsson and C. F. Westin. Normalized and differential convolution: Methods for interpolation and filtering of incomplete and uncertain data. In *CVPR*, pages 515–516, New York City, June 1993.

[10] J. A. Noble. Finding corners. *Image and Vision Computing*, 6(2):121–128, 1988.