MVA '98 IAPR Workshop on Machine Vision Applications, Nov. 17-19, 1998, Makuhari, Chiba, Japan

3—26 Gradient Estimation in Uncertain Data

Frank de Jong, Lucas J. van Vliet, Pieter P. Jonker

Pattern Recognition Group
Faculty of Applied Sciences,
Delft University of Technology.
{frankdj,lucas,pieter}@ph.tn.tudelft.nl

Keywords: edges, normalized convolution, gradient

Abstract

Detecting edges in images which are distorted by unreliable or missing data samples can be done using normalized (differential) convolution. This work presents a comparison between gradient estimation using normalized convolution and gradient estimation using normalized differential convolution with regard to speed, accuracy, and noise-sensitivity.

1 Introduction

In order to accurately determine position and size of objects within an image, accurate edge detectors are required. Much work has been done in the field of edge detectors, with different methods performing better under different assumptions. Assuming no noise in the image, one may find the edges by simply writing out the difference equation in a certain pixel neighborhood and thresholding.

Using more realistic view of the presence of noise, Marr and Hildreth [1] proposed to use the zero-crossings of the Laplacian of a Gaussian.

Canny [2] looked for the optimal edge detector for an arbitrary edge profile distorted by additive Gaussian noise, by numerical optimization of three criteria: a) Good detection (high SNR), b) Good localization, c) One response per edge. For a step edge, he found a solution which is very similar to the first derivative of a Gaussian. The edges are found at the maxima of the edge detector. He also found that criteria a) and b) are principally incompatible as the first requires a large window, and the second requires a small window.

Pitas and Venetsanopoulos [3] compared filters based on nonlinear means and order statistics and found that the dispersion edge detector performs well with several types of noise. This detector sorts the pixel values in a NxN window, and then calculates the coefficients $w_k = i_k - i_{N^2-k}$, where i_k is the k-th highest value. The w_k are weighted and summed, yielding high values for edges.

More recently, van Vliet [4] proposed to use the zero crossing of the PLUS operator (PLUS = Laplace + Second Derivative in the Gradient Direction) to detect curved edges, as it improves on the accuracy of both separate filters by one order of magnitude.

Larré and Montseney [5] found, by looking at the error distributions of the magnitude and direction of the gradient, that the direction of the gradient behaves more robustly than the magnitude, if the x and y components are assumed to have an unbiased Gaussian error distribution, $N(0, \sigma)$.

However, none of these methods take the possibility into account that information is available about the error distribution of individual pixels. Errors in the camera system, or the use of irregular sampling grids can provide a mask image which indicates for each pixel the amount of certainty one has for that particular pixel. Especially in the case of irregular sampling grids a simple 1 or 0 can indicate whether or not data is available.

In order to take advantage of these certainty masks, Knuttson and Westin [6], [8] present normalized convolution (NC) and normalized differential convolution (NDC). As the NDC generally involves a much higher computational load than the NC, this work determines the trade-off between speed and precision of gradient detection for uncertain data.

The paper is organized as follows: Section 2 will briefly review the N(D)C theory. In Section 3, both NC and NDC are applied to gradient calculation, which yields two gradient estimators.

Section 4 shows results of a number of tests for comparing the estimators. Section 5 draws conclusions based on the results.

2 Normalized (Differential) Convolution

If we consider the mathematical operations of convolution of a data set (for instance, a pixel neighborhood in an image) with a filter, we can see that for one particular data subset (neighborhood), convolution with a $W \times W$ filter window is exactly equivalent to the inner product of two $W^2 \times 1$ signal vectors.

Recalling from linear algebra that the inner product of a vector with another vector is equivalent to projecting the first vector on the second, we could view convolution as projecting on a base vector. Extending this idea, we could also project the data on a base which consists of more than one base vectors (which need not necessarily span the entire M-dimensional space).

Knowing that projecting vectors on a non-complete base will result in loss of information, we try to find the $\vec{f_b}$ that minimises (in least-squares sense) the error ϵ that we make in projection, that is:

$$\min_{(\vec{f_b})_i} \epsilon = \|\vec{f} - B\vec{f_b}\|^2 \tag{1}$$

where the $(\vec{f_b})_i$ denote the components of vector $\vec{f}$ on the base $\mathcal{B}$, and B denotes the matrix formed by putting the base vectors of $\mathcal{B}$ in the columns. By taking the derivative of ϵ with respect to the f_b and equating them to zero, we find:

$$B^*\vec{f} = B^*B\vec{f_b} \tag{2}$$

where B^* denotes the adjoint of B. Note that B^*B is an $N \times N$ matrix, where N denotes the number of base vectors in $\mathcal{B}$. This means that it's invertible so long as the base vectors are not linearly dependent:

$$\vec{f_b} = (B^*B)^{-1}B^*\vec{f} \tag{3}$$

Now that we've found the optimal way to map the data onto a basis, we'd like to introduce a windowing function, so that we can progressively lower the influence of pixels that are farther from the current pixel's neighborhood.

Westin shows that Eq. 3 still holds if the base vectors $\vec{b_i}$ in the columns of B are multiplied by a scalar function, provided that this function (Westin calls it an applicability function, a) does not introduce dependencies between the base vectors. For most bases, this requirement is easily met.

By symmetry, the same procedure can be applied to an applicability function for the data, which indicates how valid the data is for each data point. Westin calls this the certainty function, c. Then, the complete formula for normalized convolution is:

$$\begin{aligned} \vec{f_b} &= G^{-1}((AB^*)C\vec{f}) \\ G &= (AB^*)CB \end{aligned} \tag{4}$$

where A is a diagnonal matrix corresponding to the function a, and C is a diagonal matrix corresponding to c.

In projecting $\vec{f}$ on base vectors $\vec{b_i}$ we have effectively fitted a linear system, $\vec{f} = \sum_i (\vec{f_b})_i \vec{b_i}$. Now suppose we have a constant term β in this equation, which we are not interested in: $\vec{f} = \beta + \sum_i (\vec{f_b})_i \vec{b_i}$. We could then simply subtract components k from component 1, yielding a 'differential' vector $\vec{\delta}$, with components:

$$\delta_k = (f_k - f_1) = \alpha(b_{ik} - b_{i1}) \tag{5}$$

where b_{ik} denotes the k-th component of the i-th base vector and β has been eliminated. But there is no particular reason why we chose to combine $(k, 1)$ component pairs. In fact, we could have taken any combination of components, say $(k, 2)$, or better still: all combinations (k, l).

Returning to NC, and writing Eq 4 out in a sum, the numerator is a vector with components $n_i = \sum_k a_k c_k b_{ik} f_k$. We see the applicability and certainty values a_k and c_k, which we would also like in the differential form. Taking simply the product of the a_k and a_l of the components k and l we compare, (and similar for the c_k and c_l), we come to:

$$d_i = \sum_{kl} a_k a_l c_k c_l (b_{ik} - b_{il})(f_l - f_k) \tag{6}$$

The double sum can be rewritten into four single sums, using inner product notation:

$$d_i = \langle a, c\rangle\langle a\vec{b_i}, c\vec{f}\rangle \; - \langle a\vec{b_i}, c\rangle\langle a, c\vec{f}\rangle \tag{7}$$

As with NC in Eq 4, we now normalize this number by multiplying with the inverse of a matrix G, whose components are formed by substituting the data vector by the base vectors $\vec{b_j}$, yielding:

$$G_{ij} = \langle a, c\rangle\langle a\vec{b_i}, c\vec{b_j}\rangle \; - \langle a\vec{b_i}, c\rangle\langle a, c\vec{b_j}\rangle \tag{8}$$

The components of f on $\mathcal{B}$ are then:

$$\vec{f_b} = G^{-1}\vec{d} \tag{9}$$

Eqs 7 - 9 define normalized differential convolution.

3 Estimation of Gradients

We now apply the base-projection filtering paradigm on gradient estimation. In finding the gradient, we are interested in the local slope of the data points. Therefore, we project the data on base vectors describing a ramp in the x direction: $\mathcal{B} = \{\vec{b_1}, \vec{b_2}\} = \{1, x\}$. If we use NDC, the constant term is removed (as was described above), so we need only project on x. Projecting on x only, we get:

$$f_b = \frac{\langle g, c\rangle\langle g\vec{x}, c\vec{f}\rangle \; - \langle g\vec{x}, c\rangle\langle g, c\vec{f}\rangle}{\langle g, c\rangle\langle g\vec{x}, c\vec{x}\rangle \; - \langle g\vec{x}, c\rangle\langle g, c\vec{x}\rangle} \tag{10}$$

Differentiating the Gaussian, $g_x = -\frac{x}{\sigma^2} g$, and inserting this into Eq 10, we get:

$$f_b = \frac{\langle g, c\rangle\langle g_x, c\vec{f}\rangle \; - \langle g_x, c\rangle\langle g, c\vec{f}\rangle}{\langle g, c\rangle\langle g_x, c\vec{x}\rangle \; + \sigma^2\langle g_x, c\rangle^2} \tag{11}$$

We now focus our attention on a gradient estimator which, as we will see, resembles NDC in a way. The estimator, which we shall call DoNC (Derivative-of-NC) is defined by:

$$DoNC = \frac{\partial}{\partial x}\left(\frac{c\vec{f} \otimes g}{c \otimes g}\right) \tag{12}$$

in which $\otimes$ denotes convolution. The mathematical operations are equivalent to NC mapping onto one base vector $\vec{b_1} = (1, 1, ..., 1)$ (which means interpolation), using c as a certainty function and a Gaussian g as applicability function. Writing out Eq 12 analytically, we get

$$DoNC = \frac{(c \otimes g)(cf \otimes g_x) - (c \otimes g_x)(cf \otimes g)}{(c \otimes g)^2} \tag{13}$$

We expect the NDC method to perform better under noisy conditions because it fits a sloping plane, whereas the DoNC estimator interpolates, and then takes a derivative. It is interesting to note that Eq 13 and Eq 10 have identical numerators.

For industrial applications, speed is also an issue. As both methods only use convolutions with a Gaussian and its derivative, the filters can be relatively fast because those convolutions are separable in the x and y direction. Note that projecting on more than 1 base vectors would require matrix inversions.

For reference, we have also included a combination the SUSAN smoothing method with Derivative-of-Gaussian gradient filtering. The SUSAN method was reported by Smith [7] to perform best among a set of non-linear smoothers. Note that the SUSAN method was not designed for the type of noise we are subjecting our data to. It does a weighted interpolation in a neighborhood, in which not only the distance between pixels, but also the difference in grey value is weighted:

$$J(\vec{r_0}) = \frac{\sum_{\vec{r} \neq \vec{r_0}} I(\vec{r}) e^{-\frac{d^2}{2\sigma^2} - \left(\frac{(I(\vec{r}) - I(\vec{r_0}))}{t}\right)^2}}{\sum_{\vec{r} \neq \vec{r_0}} e^{-\frac{d^2}{2\sigma^2} - \left(\frac{(I(\vec{r}) - I(\vec{r_0}))}{t}\right)^2}} \tag{14}$$

where d^2 denotes the distance $\|\vec{r} - \vec{r_0}\|$, $\vec{r_0}$ is the center pixel, t is an intensity threshold and σ determines the window width.

4 Experimental Results

The filters were applied to 128x128 pixels, 8-bit images containing: a) a sinusoidal wave, b) a block pattern, c) the well-known "Lena" image. The images were corrupted by a) progressive random data removal, b) 10% data removal with progressive additive Gaussian noise, and c) 50% data removal with progressive additive Gaussian noise. The resulting gradient estimates were compared with a 'ground truth' gradient, for which we took the exact solution of the gradient for the sinusoid image, and a Derivative-of-Gaussian (DoG) with support of $\sigma = 1.5$ for the block pattern and the Lena image. The Gaussians in the NDC and DoNC filters were taken with the same σ value. In the random data removal experiments, the removal masks were used as certainty functions for the NDC and DoNC methods.

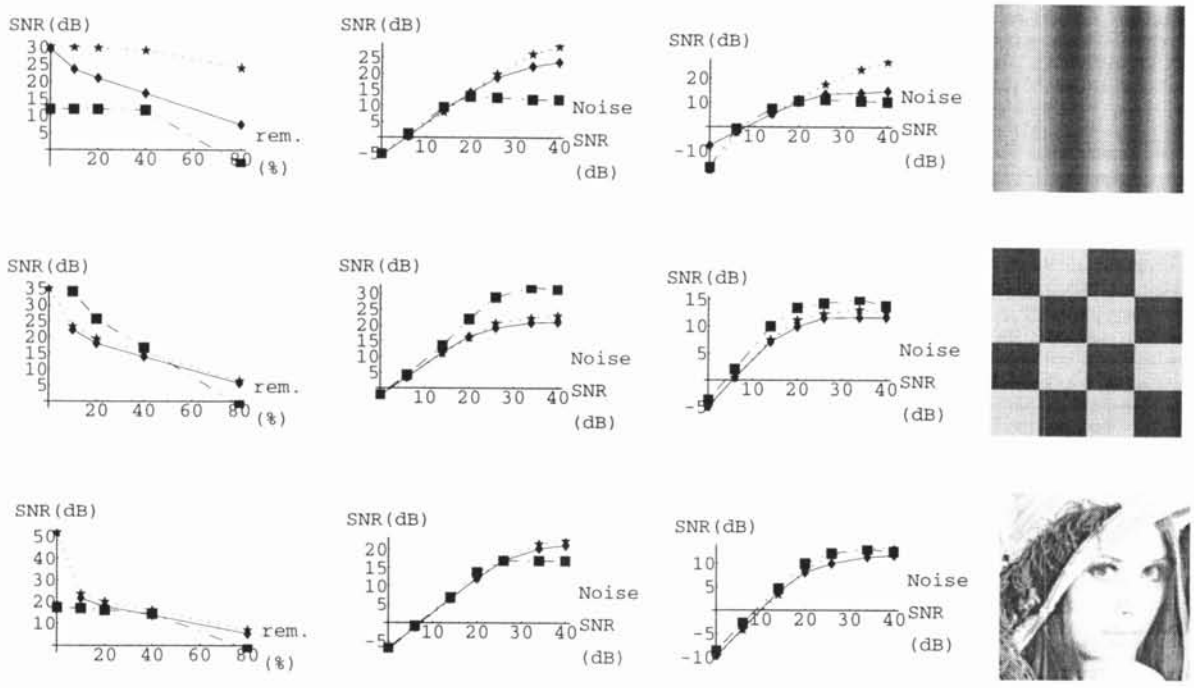
The sinusoidal wave image has an amplitude 50 around a center grey value 128, and a frequency of about 40 pixels per period. The block pattern consisted of interleaved 32x32 pixel blocks of pixels with greyvalues 78 and 178.

The NDC and DoNC methods performed within the same speed order of magnitude. On a Sun Ultra 10/300, a 256x256 image took about 0.9 seconds. The SUSAN method was about 7 times slower than the NDC and DoNC.

The results of the experiments concerning accuracy are combined in Figure 1. Graphs in the same row correspond to the same source image, graphs in the same column correspond to similar noise conditions.

The NDC performed best for the slowly varying gradient in the sinusoidal image and for the Lena image. However, it may be noted that all three methods behave similarly for additive Gaussian noise above 20 dB SNR.

For the block pattern, the Derivative-of-SUSAN method performs best because the SUSAN filter is good at restoring constant greyvalue areas. The derivative which is subsequently calculated, is exactly the same as the ground truth method, yielding a small difference between the two.



(a) Data removal (b) 10% rem + Gaussian (c) 50% rem + Gaussian (d) images

Figure 1: Signal-to-noise ratio of the estimated gradient relative to the ground truth gradient, for different combinations of noise. The estimators are differential-of-NC (diamonds) , NDC (stars), differential-of-SUSAN (squares).

It is interesting to note that the NDC and DoNC still have high SNR when 20-40% of the data is removed, even when a non-synthetic image is used.

5 Conclusion

In situations where information about the trustworthiness (or even absence) of data is available on a per-pixel basis, normalized differential convolution can be used to estimate the gradient in a robust way. By using Gaussians and Derivative-of-Gaussians for the base vectors, a separable filter can be constructed which allows relatively fast evaluation of local gradients. The DoNC filter, which is similar to NDC, can be constructed by differentiation of the NC equation; its accuracy shows similar behavior, but it has a lower computational complexity. Both methods yield approximately the same results for Gaussian noise worse than 20 dB SNR.

References

[1] D.C. Marr and E. Hildreth, "Theory of edge detection", Proc. Royal Society London, vol. B 207, pp. 187-217, 1980

[2] J.F. Canny, "A computational approach to edge detection", IEEE Transactions on PAMI, vol PAMI-8, pp. 679-698, 1986

[3] I. Pitas and A.N. Venetsanopoulos, "Edge Detectors Based on Nonlinear Filters", IEEE Transactions on PAMI, vol PAMI-8, pp. 539-550, 1986

[4] L.J. van Vliet, "Grey-Scale Measurements in Multi- Dimensional Digitized Images", PhD-thesis, Delft University Press, ISBN 90-6275-904-1/CIP, 1993

[5] A. Larré and E. Montseny, "A Step Edge Detector Algorithm Based on Symbolic Analysis", Proc. Int. Conf. on Pattern Recognition, vol I, pp 6-11, 1994, Jerusalem

[6] C-F. Westin, "A Tensor Framework for Multidimensional Signal Processing", PhD-thesis, Linköping University, ISBN 91-7871-421-4, 1994

[7] S.M. Smith, "Flexible Filter Neighborhood Designation", Proc. Int. Conf. on Pattern Recognition, vol I, pp 206-212, 1996, Vienna

[8] H. Knuttson and C-F. Westin, "Normalized Convolution: A Technique for Filtering Incomplete and Uncertain data", Proc. 8th Scand. Conf. on Image Analysis, Tronsø(Norway), 1993, pp 997-1006.