Knutsson H, Westin CF. Normalized and differential convolution: Methods for interpolation and filtering of incomplete and uncertain data. CVPR'93. New York City, USA, 1993;515-523.

Normalized and Differential Convolution

Methods for Interpolation and Filtering of Incomplete and Uncertain Data

Hans Knutsson Carl-Fredrik Westin

Computer Vision Laboratory
Department of Electrical Engineering
Linköping University, S-581 83 Linköping, Sweden
Fax: +46 13 138526, email: knutte@isy.liu.se, email: westin@isy.liu.se

Abstract

In this paper it is shown how false operator responses due to missing or uncertain data can be significantly reduced or eliminated. Perhaps the most well-known of such effects are the various 'edge effects' which invariably occur at the edges of the input data set. Further, it is shown how operators having a higher degree of selectivity and higher tolerance against noise can be constructed using simple combinations of appropriately chosen convolutions. The theory is based on linear operations and is general in that it allows for both data and operators to be scalars, vectors or tensors of higher order.

Three new methods are presented: Normalized convolution, Differential convolution *and* Normalized Differential convolution. *All three methods are examples of the power of the* signal/certainty - *philosophy, i.e. the separation of both data and operator into a signal part and a certainty part. Missing data is simply handled by setting the certainty to zero. In the case of uncertain data, an estimate of the certainty must accompany the data. Localization or 'windowing' of operators is done using an applicability function, the operator equivalent to certainty, not by changing the actual operator coefficients. Spatially or temporally limited operators are handled by setting the applicability function to zero outside the window.*

Consistent with the philosophy of this paper all algorithms produce a certainty estimate to be used if further processing is needed. Spectrum analysis is discussed and examples of the performance of gradient, divergence and curl operators are given.

1 Introduction

Information representation is an important issue in all multi-level signal processing systems. The issue is complex and what is a good information representation varies with the application. Nevertheless, we stress that an important common feature for a good representation is that it should keep statement and certainty of statement apart,

From a philosophical point of view it should be no argument that 'knowing' and 'not knowing' are different situations regardless of what can be known. Thoughts along these lines are of course by no means new ([11] [3] [15] [5]) and can, dependent on point of view, be said to have relations to both probability theory, fuzzy set theory, quantum mechanics and evidence calculus. However, it is felt that the vision community would benefit from an increased awareness of the importance of these ideas. The present paper is intended to be a contribution towards this end.

Consider the following simple example. Vectors are commonly used for representing speed and direction of speed. How should a vector having zero magnitude be interpreted? Is the speed zero or do we have the case that no information about the velocity is available? Note that information of local image velocity is impossible to recover in flat regions. In an example below it is shown that, using a vector as the sole representation for local velocity, borders between regions of missing data and good data can induce strong erratic responses.

In practice, having, or being able to produce, additional certainty information is not unusual, e.g. range data normally consists of two parts; a scalar value defining the distance and an energy measure basically defining points, where the range camera has failed to estimate the distance, so called drop-outs.

2 Notations

Before presenting the concepts of *normalized* and *differential* convolution we will define the notations that will be used in this paper.

$\boldsymbol{\xi}$	is the global spatial coordinate.
$\mathbf{x}$	is the local spatial coordinate.
$\mathbf{T}(\boldsymbol{\xi})$	is a **tensor** representing the input signal.
$c(\boldsymbol{\xi})$	is a positive scalar function representing the **certainty** of $\mathbf{T}(\boldsymbol{\xi})$.
$\mathbf{B}(\mathbf{x})$	is a tensor representing the operator filter **basis**.
$a(\mathbf{x})$	is the operator equivalent to certainty, a positive scalar function representing the **applicability** of $\mathbf{B}(\mathbf{x})$.

The philosophy is that data as well as operators are accompanied by a scalar component representing the appropriate 'weight' to put on the data or operator values.

Definition 1 The basic operation needed for the following presentation is a generalized form of convolution which can be written:

$$\mathbf{U}(\boldsymbol{\xi}) = \sum_{\mathbf{x}} a(\mathbf{x})\mathbf{B}(\mathbf{x}) \odot c(\boldsymbol{\xi}-\mathbf{x})\mathbf{T}(\boldsymbol{\xi}-\mathbf{x}) \qquad (1)$$

where $\odot$ denotes some multilinear operation (in standard convolution this operation is scalar multiplication). As long as the basic operation is understood explicitly indicating the dependence on the spatial coordinates $\boldsymbol{\xi}$ and $\mathbf{x}$ serves no purpose. Thus, for clarity, the above expression for general convolution will in the following be written:

$$\mathbf{U} \;=\; \{a\mathbf{B} \;\widehat{\odot}\; c\mathbf{T}\} \qquad (2)$$

where the 'hat' over the multilinear operation symbol serves as a marker of the operation involved in the convolution (this is useful when more than one operation symbol appear within the brackets).

3 Normalized Convolution

Probably the most commonly used operation in image processing is convolution. The fundamental assumption underlying this fact is that the original representation of a particular neighbourhood, i.e. the values of the signal for each pixel, is not a good one but that the neighbourhood can be better understood when expressed in terms of a set of carefully chosen new basis functions. The original basis, i.e. the set of impulses located at each pixel, is considered to be fixed, and thus the change of base becomes trivial. However, the assumption that the data is represented in a fixed basis is often false and neglecting to note this fact can introduce severe errors.

The most prominent example of a 'changing basis' situation is the missing samples case. A missing sample means that the impulse basis function is missing and not that the coordinate, i.e. the value at that position, is zero. In a changing basis situation proper signal analysis is possible only if the representation of the signal is complete, i.e. the representation includes not only the coordinates of the signal but also the basis in which the coordinates are given. For images a general representation of one basis impulse function would consist of its spatial position and of its strength. In the following spatial positions are assumed to be quantized in a regular fashion so that positions can be identified by simple enumeration. The impulse strength, however, will be explicitly represented and denoted by c. In the following c will be referred to as being the *certainty* of the signal.

Normalized convolution is a method for performing general convolution operations on data of signal/certainty type. The definition allows for the signal to be represented by tensors (i.e. scalars, vectors or tensors of higher order).

Definition 2 Let normalized convolution of $a\mathbf{B}$ and $c\mathbf{T}$ be defined and denoted by:

$$\mathbf{U}_N \;=\; \{a\mathbf{B} \;\widehat{\odot}\; c\mathbf{T}\}_N \;=\; \mathbf{N}^{-1}\mathbf{D} \qquad (3)$$

where:

$$\mathbf{D} \;=\; \{a\mathbf{B} \;\widehat{\odot}\; c\mathbf{T}\}$$

$$\mathbf{N} \;=\; \{a\mathbf{B}\odot\mathbf{B}^* \,\widehat{\cdot}\, c\}$$

The star, *, denotes complex conjugate

The multilinear operator, $\odot$, used to produce $\mathbf{D}$ and $\mathbf{N}$ is the same, i.e. summation (if any) is performed over corresponding indices. It should be noted that $\mathbf{N}$ in equation (3) contains a description of the certainties associated with the new basis functions and can be used if further processing is needed.

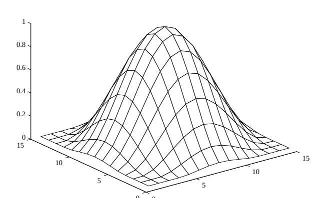

Figure 1: *Example of an applicability function $\alpha = 0$, $\beta = 2$, $r_{max} = 8$*

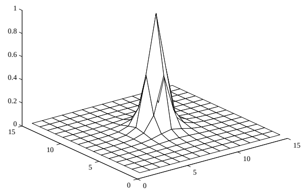

Figure 2: *An applicability function having $\alpha = 3$, $\beta = 0$, $r_{max} = 8$*

The applicability function can be said to define the localization of the convolution operator. The appropriate choice of this function depends of course on the application. The family of applicability functions used in our experiments is given by:

$$a \;=\; \begin{cases} r^{-\alpha}\; cos^{\beta}(\frac{\pi\; r}{2\; r_{max}}) & r < r_{max} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where:

r denotes the distance from the neighbourhood center.

α and β are positive integers

3.1 Mean square optimality

Normalized convolution produces a description of the neighbourhood which is optimal in the mean square sense. To see this, express a given neigbourhood, $\mathbf{t}$, in a set of basis functions given by a matrix B and the coefficients $\mathbf{u}$. (In this section standard matrix and vector notation will be used.)

$$\mathbf{t}' \;=\; B\mathbf{u} \tag{5}$$

Normally $\mathbf{u}$ will be of lower dimensionality than $\mathbf{t}$ and the neigbourhood can not be described exactly, i.e. $\mathbf{t}' \neq \mathbf{t}$. It is well known, however, that, for a given set of basis functions,B , the mean square error, $\|\mathbf{t}' - \mathbf{t}\|$, is minimized by choosing $\mathbf{u}$ to be:

$$\mathbf{u} \;=\; [B^T B]^{-1} B^T \mathbf{t} \tag{6}$$

A weighted mean square solution can be obtained by introducing a diagonal matrix, W.

$$W\mathbf{t}' \;=\; WB\mathbf{u} \tag{7}$$

Then the minimum of $\|W(\mathbf{t}' - \mathbf{t})\|$ is obtained by choosing $\mathbf{u}$ to be:

$$\mathbf{u} \;=\; [(WB)^T WB]^{-1} (WB)^T W\mathbf{t} \tag{8}$$

Figure 3: **Top left:** *The famous Lena-image has been degraded to a grey-level test image only containing 10% of the original information.* **Top right:** *Interpolation using standard convolution with a normalized smoothing filter (see figure 1).* **Bottom left:** *Interpolation using normalized convolution with the same filter as applicability function.* **Bottom right:** *Normalized convolution using a more local applicability function (see figure 2).*

which can be rewritten and split into two parts, $\mathbf{N}^{-1}$ and $\mathbf{D}$.

$$\mathbf{u} \;=\; \underbrace{[B^T W^2 B]}_{\mathbf{N}}{}^{-1}\;\; \underbrace{B^T W^2 \mathbf{t}}_{\mathbf{D}} \tag{9}$$

After some playing around with indices it can be shown that $\mathbf{N}$ and $\mathbf{D}$ are identical to the corresponding quantities used in normalized convolution, equation (3). In normalized convolution the diagonal weighting matrix is, for a neigbourhood centered on $\boldsymbol{\xi}_0$, given by:

$$W_{ii}^2(\boldsymbol{\xi}_0) \;=\; a(\mathbf{x}_i)\; c(\boldsymbol{\xi}_0 - \mathbf{x}_i) \tag{10}$$

Thus, normalized convolution can be seen as a method for obtaining a local weighted mean square error description of the input signal. The input signal is described in terms of the basis function set, B, the weights are adaptive and given by the data certainties and the operator applicability

Figure 4: **Left**: *Test image.* **Right**: *Interpolation using normalized convolution.*

3.2 0:th order interpolation

An illustrative example is constituted by the use of normalized convolution to obtain an efficient interpolation algorithm in a missing sample situation. In the simplest possible case the operator filter basis consists of only one position invariant basis function, i.e. $\mathbf{B} = 1$. The interpolated result can then be expressed as:

$$\mathbf{U}_N \;=\; \{a \mathbin{\hat{\cdot}} c\mathbf{T}\}_N \;=\; \{a \mathbin{\hat{\cdot}} c\}^{-1}\{a \mathbin{\hat{\cdot}} c\mathbf{T}\} \qquad (11)$$

where $\cdot$ denotes standard scalar multiplication

Since $\mathbf{B}$ now is a constant, i.e. a 0:th order tensor it is evident that all components of $\mathbf{T}$ will be subject to the same transformation and it is consequently enough to study the case where $\mathbf{T}$ is a scalar g (for gray scale). Equation (11) then reduces to the following form:

$$u_N \;=\; \{a \mathbin{\hat{\cdot}} cg\}_N \;=\; \{a \mathbin{\hat{\cdot}} c\}^{-1}\{a \mathbin{\hat{\cdot}} cg\} \qquad (12)$$

where:

cg is the certainty value times the grey-level value

It may be worth mentioning that if all certainties are equal, equation (12) is reduced to standard convolution and the standard representation of a grey-level image:

$$\{a \mathbin{\hat{\cdot}} cg\}_N = \{a \mathbin{\hat{\cdot}} c\}^{-1}\{a \mathbin{\hat{\cdot}} cg\} = \frac{c\{a \mathbin{\hat{\cdot}} g\}}{c\{a \mathbin{\hat{\cdot}} 1\}} = \{a' \mathbin{\hat{\cdot}} g\} \qquad (13)$$

where a' is the normalized applicability function. In other words, interpolation by convolution using a kernel maintaining the local DC-component, a technique widely used when resampling images to a larger size.

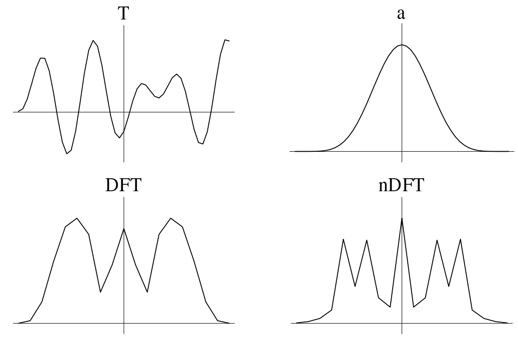

Figure 5: *A spectrum analysis experiment. The top left shows the signal. The top right shows the function that was used both as window and as applicability function. At the bottom the result of standard windowed DFT analysis (left) and of spectrum analysis using normalized convolution (right) is shown.*

Experimental results

The first example is interpolation of a sparsely *irregular* sampled test-image. The image has been constructed by the use of gated white noise. The threshold was chosen so only 10% of the data remained, see figure 3 (top left). An attempt to reconstruct this image by simple smoothing is of course deemed to fail due to the sample density variation. The result of the smoothing operation using the filter shown in figure 1 is shown in figure 3 (top right).

The result of the normalized convolution using the same filter as applicability function is shown in figure 3 (bottom left). The local normalization performed by normalized convolution compensates effectively for the sample density variations. Although the result is satisfactory, we will show that an even better result can be obtained using a more localized applicability function. There is no need for smoothing the image since the shape of the applicability function is compensated for in the normalization. The parameter values for the applicability function used to produce the result shown in figure 3 (bottom right) were $\alpha = 3$ and $\beta = 2$. Using such a filter in normalized convolutions results in an adaptive smoothing of the reconstructed image. The more missing samples the more smoothing is performed.

The second example is interpolation of a densely sampled image having large missing regions, see figure 4. The result shows that the algorithm is capable of filling in the 'holes' with 'plausible' data. The test also shows that the algorithm performs well close to the image border since this region of an image also belongs

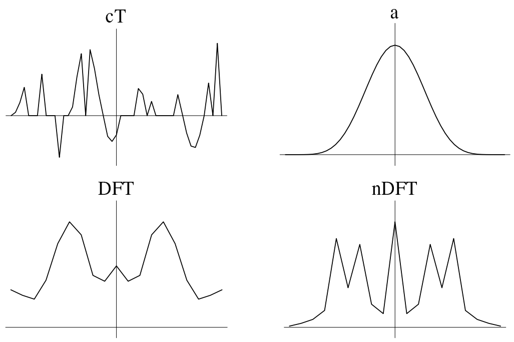

Figure 6: *The result of an experiment using the same signal as in figure 5, the difference being that 50% of the samples were removed at random. The removal of the samples has a dramatic effect on the standard DFT of the signal, the spectrum analysis performed by normalized convolution is, however, robust.*

to the case densely samples/large missing regions.

3.3 Spectrum analysis

Spectrum analysis provides a good example of how normalized convolution can be used. All spectrum analysis methods, explicitly or implicitly, involve some kind of windowing operation. In image processing the windows are typically small and the standard windowing operation is multiplication, i.e. the signal is simply multiplied with the window function prior to the analysis. The effect is that what is being analyzed is not the signal but a corrupted version thereof, the implications of which are well known and unwanted. This way of solving the locality problem is a clear violation of the signal/certainty principle.

A much preferable way of attaining local signal analysis is to let the signal be accompanied by a value stating how important the signal, at a given point, is for the analysis. In this way the signal can be left unchanged and locality be introduced by letting the importance the signal decrease with the distance from the center according to a window function. This is precisely what normalized convolution is capable of, the applicability function being the equivalent of a windowing function.

In a missing sample situation the advantages of the signal/certainty approach become even more apparent. Standard spectrum analysis has no way of coping with this situation, but the performance of normalized convolution is highly robust.

Experimental results

Figure 5 shows a spectrum analysis experiment. The top left shows the signal which is a sum of a constant term plus two different sinusoids. The top right shows the function that was used both as window and as applicability function. The bottom of figure 5 shows the result of standard windowed DFT analysis (left) and of spectrum analysis using normalized convolution (right). The advantage of using normalized convolution is evident - the frequency domain resolution is significantly improved and the two frequency components of the sinusoids are clearly separated.

Figure 6 shows the result of an experiment using the same signal as in figure 5, the difference being that 50% of the samples were removed at random. Unsurprisingly the removal of the samples has a dramatic effect on the standard DFT of the signal. The spectrum analysis performed by normalized convolution is, however, remarkably robust and only minor changes can be seen.

3.4 Signal reconstruction

Another application where normalized convolution can be applied is in signal reconstruction. Signal reconstruction always implies the use of a model for the signal. In the normalized convolution case the model is implicit in the chosen basis functions $\mathbf{B}$. It can be shown that the signal reconstruction performed by normalized convolution is a weighted least squares solution using the available basis functions, [7]. However, a good reconstruction can only be hoped for if the chosen basis functions are likely to be able to account for a large part of the signal.

Experimental results

Figure 7 shows a signal reconstruction experiment where 32 out of 50 samples were missing. The figure shows the signal (top left), the signal certainty product (top right), the distribution over the used basis functions (bottom left) and the reconstructed signal (bottom right). The basis functions consisted of a constant and 5 sin/cos pairs having frequencies 0.5, 1.0, 1.5, 2.0 and 2.5 cycles per graph width. Although the signal was chosen not to fit any of the individual basis functions, the reconstruction of the signal is good.

3.5 Higher order interpolation

In section 3.2, interpolation using *one* basis function was discussed. In this section it will be shown that even better interpolation results can be achieved using a larger set of basis functions. The reconstruction of the ramp function in the previous section (figure 7) was based on inverting the normalized DFT, i.e.

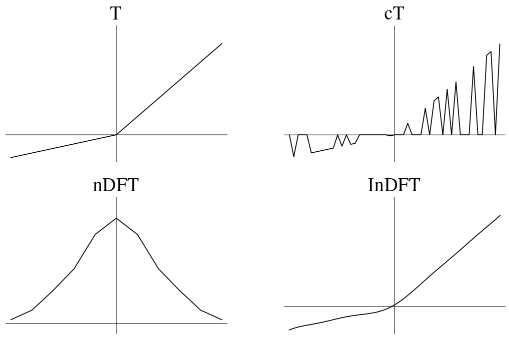

Figure 7: *A signal reconstruction experiment. The figure shows the signal (top left), the signal certainty product (top right), the distribution over the used basis functions (bottom left) and the reconstructed signal (bottom right).*

based on one measurement centered on the signal. In this section convolution is performed as in section 3.2, i.e. each part of the signal is reconstructed locally.

Experimental results

In all three examples below, seven basis functions were used; one DC function and three sin/cos pairs having one, two and three cycles per graph width.

Figure 8 shows higher order interpolation of a "random walk" signal. This signal has a spectrum decreasing as one over the frequency variable (as the spectrum in many images). Although more than halves the samples was removed at random, the reconstruction is good.

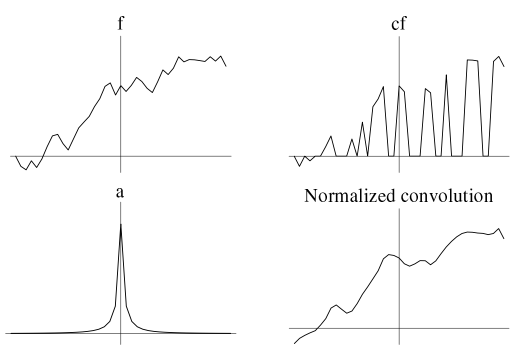

Figure 8: *Interpolation example using normalized convolution. The figure shows the signal (top left), the signal certainty product (top right), the used applicability function (bottom left) and the reconstructed signal (bottom right).*

Figure 9 shows higher order interpolation of a smoothly varying signal. In this case, when the used basis functions well describe the signal, it is almost completely restored although large parts is missing.

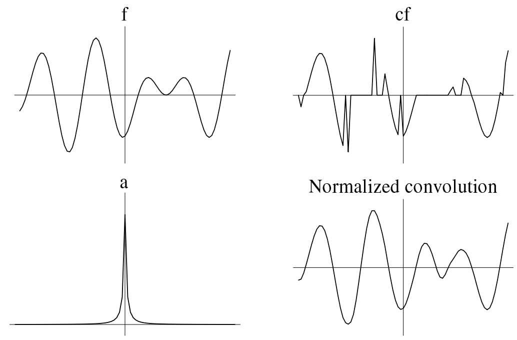

Figure 9: *Interpolation example using normalized convolution. The figure shows the signal (top left), the signal certainty product (top right), the used applicability function (bottom left) and the reconstructed signal (bottom right).*

Figure 10 shows higher order interpolation of a a signal containing the signal presented in figure 9 plus a term varying very fast. In areas where the certainties is high, the reconstruction is very good. In areas where large parts are missing, a "low-pass" version of the signal is reconstructed. The signal is adaptively smoothed.

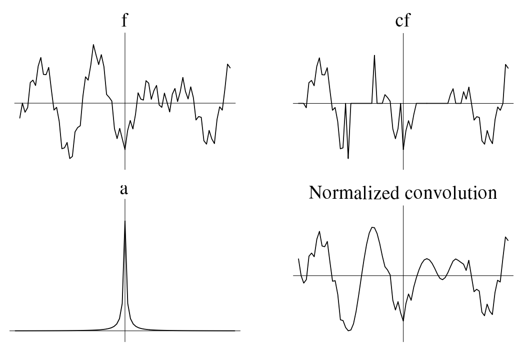

Figure 10: *Interpolation example using normalized convolution. The figure shows the signal (top left), the signal certainty product (top right), the used applicability function (bottom left) and the reconstructed signal (bottom right).*

4 Differential Convolution

In this section we will discuss an operation termed *differential convolution.* This operation can be shown to be equivalent to locally weighted sums over all operator differences acting on the corresponding data differences - hence the name. This description may, to begin with, not give a full understanding of the operation. The key words are, however, *operator differences* and *data differences*. This makes the operation insensitive to any constant term in the input signal.

Definition 3 Let differential convolution between $a\mathbf{B}$ and $c\mathbf{T}$ be defined and denoted by:

$$\begin{aligned} \mathbf{U}_\Delta &= \{a\mathbf{B} \,\widehat{\odot}\, c\mathbf{T}\}_\Delta = \\ &\{a \,\hat{\cdot}\, c\}\{a\mathbf{B} \,\widehat{\odot}\, c\mathbf{T}\} - \{a\mathbf{B} \,\hat{\cdot}\, c\} \odot \{a \,\hat{\cdot}\, c\mathbf{T}\} \end{aligned} \tag{14}$$

In definition 3, it is seen that differential convolution is based on a nonlinear combination of different standard convolutions. The term, $\{a\ c\}$, can be regarded as the local certainty energy and the second term, $\{a\mathbf{B}\ c\mathbf{T}\}$, is the term that corresponds to standard convolution. When it comes to the third and fourth term the interpretation is somewhat harder. Each operator in the filter is weighted locally with corresponding data producing a weighted average operator where the weights are given by the data.

$$\{a\mathbf{B} \,\hat{\cdot}\, c\} \leftrightarrow \text{data dependent mean-operator}$$

For the fourth term it is vice versa. The mean-data is calculated using the operator certainty as weights.

$$\{a \,\hat{\cdot}\, c\mathbf{T}\} \leftrightarrow \text{operator dependent mean-data}$$

Differential convolution should consequently be interpreted as: A standard convolution weighted with the local energy minus the "mean" operator acting on the "mean" data. It has been shown, [6], that differential convolution performs summation over all operator differences acting on corresponding data differences, i.e.:

$$\begin{aligned} \mathbf{U}_\Delta &= \{a\mathbf{B} \,\widehat{\odot}\, c\mathbf{T}\}_\Delta = \\ &\frac{1}{2}\sum_{ij} a_i a_j c_i c_j (\mathbf{B}_i - \mathbf{B}_j)(\mathbf{T}_i - \mathbf{T}_j) \end{aligned} \tag{15}$$

It may be worth repeating that the double sum in equation (15) is never carried out, the result is achieved by the combination of four simple sums. Note that only point pairs having non zero a_i, c_i, a_j and c_j will contribute to the sum. If $\mathbf{B}$ or $\mathbf{T}$ is constant, the expression will be sum to zero.

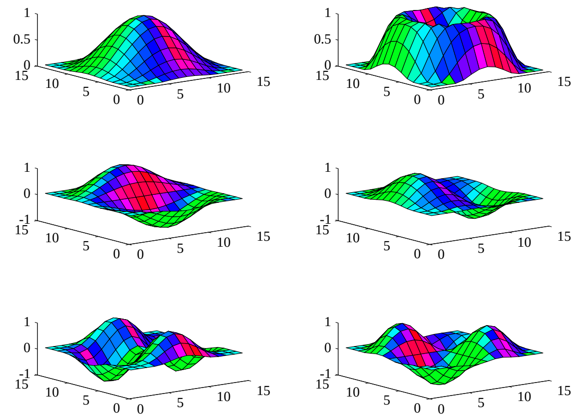

Figure 11: *Filters shown are the applicability functions times operator components used for gradient estimation. Certainty kernel (top:left) as defined above having $\tau = 0, \beta = 2, r_{max} = 8$ Operator kernels values are $a, a(x^2 + y^2), ax, ay, a(x^2 - y^2), a2xy$*

5 Normalized Differential Convolution

Normalized Differential convolution is, as the name indicates, a combination of the first presented concepts in this paper, Normalized convolution, and Differential convolution. Normalized differential convolution are methods for performing general differential operations on data of signal/certainty type in cases where the DC-component of the output data is zero or uninteresting.

We begin with the definition:

Definition 4 Let normalized differential convolution between $a\mathbf{B}$ and $c\mathbf{T}$ be defined and denoted by:

$$\mathbf{U}_{N\Delta} = \{a\mathbf{B} \,\widehat{\odot}\, \mathbf{T}\}_{N\Delta} = \mathbf{N}_\Delta^{-1}\mathbf{D}_\Delta \tag{16}$$

where:

$$\begin{aligned} \mathbf{D}_\Delta &= \{a \,\hat{\cdot}\, c\}\{a\mathbf{B} \,\widehat{\odot}\, c\mathbf{T}\} - \{a\mathbf{B} \,\hat{\cdot}\, c\} \odot \{a \,\hat{\cdot}\, c\mathbf{T}\} \\ \mathbf{N}_\Delta &= \{a \,\hat{\cdot}\, c\}\{a\mathbf{B}\odot\mathbf{B}^* \,\hat{\cdot}\, c\} - \\ &\qquad \{a\mathbf{B} \,\hat{\cdot}\, c\} \odot \{a\mathbf{B}^* \,\hat{\cdot}\, c\} \end{aligned}$$

5.1 Gradients from plane basis functions

As mentioned, normalized differential convolution is a useful method for performing general differential operations with an operator giving a zero or uninteresting DC component output. The example we will present in this paper is gradient estimation.

The minimum number of basis functions needed for estimating the local gradient using normalized differential convolution is one for each dimension (normalized convolution requires one more, the constant DC operator). The plane basis is, in n dimensions, given by:

$$\mathbf{B} = (x_1, x_2, \ \dots \ , x_n)$$

where x_k are the components of the spatial basis functions.

Inserting these basis functions in the definition of normalized differential convolution and using the relation derived in equation (15), writing all the indices explicitly gives:

$$\begin{aligned}
\mathbf{D}_\Delta = & \qquad (17)\\
& \{a \hat{\cdot} c\}\{a\mathbf{B} \otimes c\mathbf{T}\} - \{a\mathbf{B} \hat{\cdot} c\} \otimes \{a \hat{\cdot} c\mathbf{T}\} \qquad (18)\\
& = \frac{1}{2}\sum_{kl} \underbrace{a_k a_l c_k c_l}_{d_{kl}} \underbrace{(x_{kn} - x_{ln})}_{\Delta x_{kln}} \underbrace{(\mathbf{T}_k - \mathbf{T}_l)}_{\Delta \mathbf{T}_{kl}} \qquad (19)\\
& = \frac{1}{2}\sum_{kl} d_{kl} \Delta x_{kln} \Delta \mathbf{T}_{kl} \qquad (20)
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{N}_\Delta = & \qquad (21)\\
& \{a \hat{\cdot} c\}\{a(\mathbf{B} \otimes \mathbf{B}^*) \hat{\cdot} c\} - \{a\mathbf{B} \hat{\cdot} c\} \otimes \{a\mathbf{B}^* \hat{\cdot} c\}\\
& = \frac{1}{2}\sum_{kl} d_{kl} \Delta x_{kln} \Delta x_{klm} \qquad (22)
\end{aligned}$$

If the gradient, $\frac{\partial \mathbf{T}}{\partial x_m}$, is constant in the neighbourhood then

$$\Delta \mathbf{T}_{kl} = \Delta x_{klm} \frac{\partial \mathbf{T}}{\partial x_m} \quad \text{for all} \quad k, l \qquad (23)$$

Equation (20) then simplifies to:

$$\mathbf{D}_\Delta = \frac{\partial \mathbf{T}}{\partial x_m} \, \frac{1}{2}\sum_{kl} d_{kl} \Delta x_{kln} \Delta x_{klm} \qquad (24)$$

Equation (22) and (24) inserted in equation (4), the definition of normalized differential convolution gives:

$$\mathbf{N}_\Delta^{-1} \mathbf{D}_\Delta = \frac{\partial \mathbf{T}}{\partial x_m} \qquad (25)$$

which shows that the true gradient is estimated for an ideal input signal.

Experimental results

This example shows estimation of the local gradient in a sparsely irregular sampled two dimensional scalar field. The equations from definition 4 get the following form;

$$\begin{aligned}
\mathbf{D}_\Delta &= \{a \hat{\cdot} c\}\{a\mathbf{B} \,\hat{\otimes}\, c\mathbf{T}\} - \{a\mathbf{B} \hat{\cdot} c\}\{a \hat{\cdot} c\mathbf{T}\}\\
\mathbf{N}_\Delta &= \{a \hat{\cdot} c\}\{a(\mathbf{B} \otimes \mathbf{B}) \hat{\cdot} c\} - \{a\mathbf{B} \hat{\cdot} c\} \otimes \{a\mathbf{B} \hat{\cdot} c\}
\end{aligned}$$

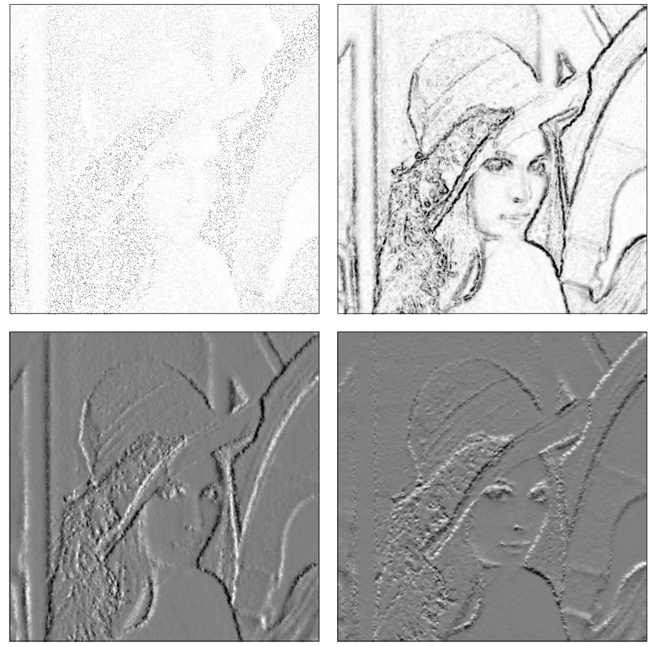

Figure 12: **Top left**: *The Lena-image degraded to 10% of the original information.* **Top right**: *Gradient magnitude estimation using normalized differential convolution.* **Bottom**: *Estimation of the x- and y-gradient in the top left image using Differential normalized convolution with a filter having $\tau = -3, \beta = 0, r_{max} = 8$.*

where $\otimes$ denotes tensor product (outer product). The basis functions needed are:

$$\mathbf{B} = (x, y) \quad \text{and} \quad \mathbf{B} \otimes \mathbf{B} = \begin{pmatrix} x^2 & xy \\ xy & y^2 \end{pmatrix} \qquad (26)$$

The filters are weighted with the applicability function a.

$$(a, ax, ay, ax^2, axy, ay^2) \qquad (27)$$

For practical purposes, an equivalent set of filters was implemented:

$$(a, ax, ay, a(x^2 + y^2), axy, a(x^2 - y^2)) \qquad (28)$$

These are shown in figure 11.

$$\begin{aligned}
\mathbf{D}_\Delta &= \{a\hat{\cdot}c\} \begin{pmatrix} \{ax\hat{\cdot}cg\} \\ \{ay\hat{\cdot}cg\} \end{pmatrix} - \begin{pmatrix} \{ax\hat{\cdot}c\} \\ \{ay\hat{\cdot}c\} \end{pmatrix} \{a\hat{\cdot}cg\}\\
\mathbf{N}_\Delta &= \{a\hat{\cdot}c\} \begin{pmatrix} \{ax^2\hat{\cdot}c\} & \{axy\hat{\cdot}c\} \\ \{axy\hat{\cdot}c\} & \{ay^2\hat{\cdot}c\} \end{pmatrix} -\\
& \quad \begin{pmatrix} \{ax\hat{\cdot}c\}\{ax\hat{\cdot}c\} & \{ax\hat{\cdot}c\}\{ay\hat{\cdot}c\} \\ \{ax\hat{\cdot}c\}\{ay\hat{\cdot}c\} & \{ay\hat{\cdot}c\}\{ay\hat{\cdot}c\} \end{pmatrix}
\end{aligned}$$

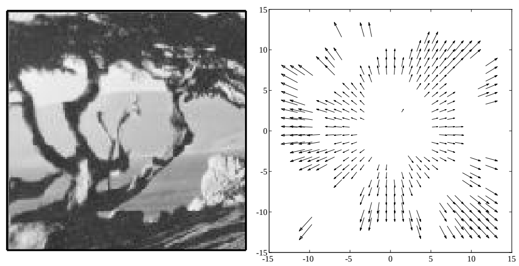

Figure 13: *Left: One frame from the tree-sequence. Right: Estimated velocities from the tree-sequence. See Landelius [10]. In order to increase the visuability the original 150x150 image has been resampled to 30x30 pixels .*

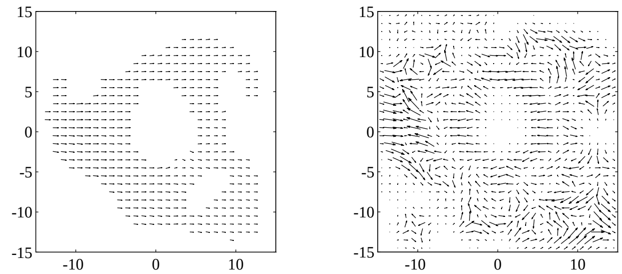

Figure 14: **Left***: Normalized differential convolution on the velocity field in figure 13 produces a field pointing to the right. If the camera had had the reversed motion. i.e. a motion away from the scene, the output field would have pointed to the left. Pure rotation of the camera would have produced a vector field pointing up or down depending on direction of rotation.* **Right***: Standard convolution. We can see that missing data completely destroy many of the estimates.*

Note that many terms in this expression are identical. The actual numbers of convolutions needed is the number of basis functions in equation (27), i.e. six scalar convolutions, see figure 11.

The sparsely sampled test image we used in the interpolation example in section 3 shall now be used for testing our new gradient estimation method. The image is filtered with the six filters shown in figure 11. Combining these outputs according to the normalized convolution procedure gives the gradient estimate, see figure 12. We can see that the algorithm has no problem coping with the large variation in the sampling density

5.2 Divergence and curl in 2D

The gradient estimation method described in the previous section can be used for estimation of first order differential invariants of image velocity fields i.e. curl, divergence and shear [9]. Since these invariants have found considerable attention in Computer Vision [1, 4, 12] it has been very exiting to test the power our signal/certainty philosophy on the problem of differentiating a velocity field. The motion field induced by a moving camera is often sparse since estimation of velocity requires moving texture or borders. The motion of all flat surfaces is unknown (not zero).

The invariants divergence, curl and shear can all be defined as a combination of partial derivatives of the image velocity field. The divergence in 2D is defined by:

$$\nabla \cdot \mathbf{T} = trace(\nabla \mathbf{T}) = T_{11} + T_{22} = \frac{\partial T_1}{\partial x_1} + \frac{\partial T_2}{\partial x_2} \quad (29)$$

The curl in 2D is a scalar defined by:

$$\nabla \times \mathbf{T} = T_{21} - T_{12} = \frac{\partial T_2}{\partial x_1} - \frac{\partial T_1}{\partial x_2} \quad (30)$$

The use of complex numbers instead of 2-dimensional vectors simplifies the formalism for the divergence and the curl operator. We therefore introduce a complex gradient operator and a complex data representation:

$$\nabla = \frac{\partial}{\partial x_1} + i\frac{\partial}{\partial x_2} \qquad \text{and} \qquad \mathbf{T} = T_1 + iT_2$$

This gives the following representation of divergence and rotation:

$$\mathrm{Div} = xT_1 + yT_2 \qquad \text{and} \qquad \mathrm{Rot} = xT_2 - yT_1$$

In polar coordinates we get $B_k = b_k\ e^{i\beta_k}$, $T_k = t_k\ e^{i\tau_k}$. The reference neighbourhood is the complex conjugate of the filter, $B_k^* = b_k\ e^{-i\beta_k}$ Inserting this in the definition of normalized differential convolution gives:

$$\mathbf{D}_\triangle \quad = \quad \{a \hat{\cdot} c\}\{a\mathbf{B} \hat{\cdot} c\mathbf{T}\} - \{a\mathbf{B} \hat{\cdot} c\}\{a \hat{\cdot} c\mathbf{T}\}$$

$$= \frac{1}{2}\sum a_k a_l c_k c_l (e^{i\beta_k} - e^{i\beta_l})(e^{i\tau_k} - e^{i\tau_l})$$
$$= 2\sum d_{kl}\sin(\frac{\Delta\beta_{kl}}{2})\sin(\frac{\Delta\tau_{kl}}{2})e^{i(\Delta\beta_{kl}+\Delta\tau_{kl})}$$

where:

$$\Delta\tau = \tau_k - \tau_l$$
$$\Delta\beta = \beta_k - \beta_l$$

and

$$\begin{aligned}\mathbf{N}_\Delta &= \{a \,\hat{\cdot}\, c\}\{a\mathbf{B}\mathbf{B}^* \,\hat{\cdot}\, c\} - \{a\mathbf{B} \,\hat{\cdot}\, c\}\{a\mathbf{B}^* \,\hat{\cdot}\, c\} \\ &= \frac{1}{2}\sum a_k a_l c_k c_l \; \|(e^{i\beta_k} - e^{i\beta_l})\|^2 \\ &= 2\sum d_{kl}\sin^2(\frac{\Delta\beta_{kl}}{2})\end{aligned}$$

Thus

$$\mathbf{N}_\Delta^{-1}\mathbf{D}_\Delta = \frac{\sum d_{kl}\sin(\frac{\Delta\beta_{kl}}{2})\sin(\frac{\Delta\tau_{kl}}{2})e^{i(\Delta\beta_{kl}+\Delta\tau_{kl})}}{\sum d_{kl}\sin^2(\frac{\Delta\beta_{kl}}{2})} \tag{31}$$

Experimental results

As a final example will be shown how local divergence and curl can be esimated in an image sequence, ([2]). In this example the camera is moving towards a flat image of a tree, (figure 13 left). The input data is the sparse estimated velocity field in figure 13 (right). The result of normalized differention convolution using a divergence/curl operator is shown in figure14(left). The result of standard convolution using the same operator is shown in figure 14(right).

6 Acknowledgement

The support from the Swedish National Board for Technical Development is gratefully acknowledged. Part of this work has been founded by the VAP project within the Esprit Basic Research Action. The authors also wish to thank the members of our computer vision group, in particular Prof. Gösta Granlund for many inspiring discussions.

References

[1] Roberto Cipolla and Andrew Blake. Surface orientation and time to contact from image divergence and deformation. In *Proceedings of ECCV–92, LNCS–Series Vol. 588*. Springer–Verlaag, 1992.

[2] David J. Fleet. *Measurement of image velocity.* Kluwer Academic Publishers, 1992. ISBN 0–7923–9198–5.

[3] G. H. Granlund and H. Knutsson. Contrast of structured and homogenous representations. In O. J. Braddick and A. C. Sleigh, editors, *Physical and Biological Processing of Images*, pages 282–303. Springer Verlag, Berlin, 1983.

[4] K. Kanatani. Structure and motion from optical flow under orthographic projection. *Computer Vision, Graphics and Image Processing*, 35:181-199, 1986.

[5] H. Knutsson. Representing local structure using tensors. In *The 6th Scandinavian Conference on Image Analysis*, pages 244–251, Oulu, Finland, June 1989. Report LiTH–ISY–I–1019, Computer Vision Laboratory, Linköping University, Sweden, 1989.

[6] H. Knutsson and C-F Westin. Differential convolution: A technique for filtering incomplete and uncertain data. In *Accepted for 8th SCIA*, Tromsø, Norway, May 1993. NOBIM.

[7] H. Knutsson and C-F Westin. Normalized and differential convolution: Methods for interpolation and filtering of incomplete and uncertain data. In *Accepted for CVPR*, New York City, USA, June 1993. IEEE.

[8] Hans Knutsson. *Filtering and Reconstruction in Image Processing.* PhD thesis, Linköping University, Sweden, 1982. Diss. No. 88.

[9] J. J. Koenderink and A. J. van Doorn. Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer. *Opt. Acta 22*, pages 773–791, 1975.

[10] T. Landelius, L. Haglund, and H. Knutsson. Depth and velocity from orientation tensor fields. In *Accepted for 8th SCIA*, Tromsø, Norway, May 1993. NOBIM.

[11] Donald M. MacKay. *Information, Mechanism and Meaning.* M.I.T. Press, Cambridge, Massachusetts and London, England, 1969.

[12] R. C. Nelson and J. Aloimonos. Using flow field divergence for obstacle avoidance: towards qualitative vision. In *Proc. 2nd Int. Conf. on Computer Vision*, pages pages 188–196, 1988.

[13] C-F Westin. Feature extraction based on a tensor image description, September 1991. Thesis No. 288, ISBN 91–7870–815–X.

[14] C-F Westin and H. Knutsson. Extraction of local symmetries using tensor field filtering. In *Proceedings of 2nd Singapore International Conference on Image Processing.* IEEE Singapore Section, September 1992.

[15] R. Wilson and H. Knutsson. Uncertainty and inference in the visual system. *IEEE Transactions on Systems, Man and Cybernetics*, 18(2), March/April 1988.

[16] R. Wilson, H. Knutsson, and G. H. Granlund. The operational definition of the position of line and edge. In *The 6th International Conference on Pattern Recognition*, Munich, Germany, October 1982.