

Optimal Difference Operator Selection

Peter Veelaert and Kristof Teelen

University College Ghent, Engineering Sciences - Ghent University Association,
Schoonmeersstraat 52, B9000 Ghent, Belgium
{Peter.Veelaert, Kristof.Teelen}@hogent.be

Abstract. Differential operators are essential in many image processing applications. Previous work has shown how to compute derivatives more accurately by examining the image locally, and by applying a difference operator which is optimal for each pixel neighborhood. The proposed technique avoids the explicit computation of fitting functions, and replaces the function fitting process by a function classification process. This paper introduces a new criterion to select the best function class and the best template size so that the optimal difference operator is applied to a given digitized function. An evaluation of the performance of the selection criterion for the computation of the Laplacian for digitized functions shows better results when compared to our previous method and the widely used Laplacian operator.

1 Introduction

Differential operators are essential in many applications in image processing and computer vision. For example, the Scale-Invariant Feature Transform (SIFT) [1] combines the Laplacian with Gaussian smoothing at different scales to extract keypoints from images. Different methods for discrete feature extraction and edge detection were proposed by Lindeberg [2], Lachaud *et al* [3], Gunn [4] and Demigny and Kamlé [5].

In fact, almost any algorithm that extracts interesting features from images, such as edges or remarkable points, relies in some way on first or higher order derivatives. Remarkably, the difference operators used to approximate a differential operator are often quite rudimentary. Furthermore, although difference operators are quite sensitive to noise, one usually assumes that Gaussian smoothing at different scales is the best way to remove noise. Previous work, however, has shown that it is possible to compute derivatives much more accurately by examining the image locally, and to apply to each pixel a difference operator which is optimal for that pixel neighborhood [6, 7, 8]. This approach gives more accurate results and needs less Gaussian blurring to obtain edges. The proposed technique avoids the explicit computation of fitting functions, a time consuming process, since it replaces the function fitting process by a function classification process.

The method proposed in this and previous papers depends heavily on the reduction of difference operators by Groebner bases [6, 7, 8]. One of the contributions of this work is that it relates difference operators to function approximation and feature detection, within the mathematical framework of ideals of difference operators. The application of Groebner bases to solve or reduce difference equations is not new, however. The

pioneering work by Oberst on the use of Groebner bases to solve systems of partial difference equations has led to considerable interest in the field [9, 10]. Gerdt *et al* use Janet-like Groebner bases to generate simple difference schemes to compute solutions for partial differential equations that involve multiple functions [11].

In previous work, we have derived an optimal difference operator for a given function class and given template size. This paper focuses on the selection of the best class and best template size for a given digitized function. We will first extend previous work by showing how to compute an optimal difference operator for each function class starting from a template with a predefined shape. Incorporating known properties of the operator has the advantage that the optimization algorithm can be sped up. We give an overview of optimal operators for different function classes and template sizes. Next, we propose a method to select the most appropriate difference operator by a criterion which involves the local features of the image and the properties of the optimal difference operator. The proposed selection criterion is illustrated by the computation of a digitized difference operator for the Laplacian. We show how the accuracy of the computation of the Laplacian can benefit from locally applying the optimal version of the Laplacian difference operator.

In section 2, we present a mathematical framework to describe the classes of fitting functions. Next, we propose a general procedure for the computation of all difference operators of a predefined shape for a function class in section 3. The procedure for finding the optimal difference operator is also described in that section. In section 4, we define a criterion for the selection of the optimal operator in accordance with the local image properties. Finally, we conclude this paper in Section 5.

2 Function Classes

To find the derivative of a discrete function f , a standard approach is to approximate f by a continuous function \tilde{g} , before taking derivatives. This is a time consuming process, which involves unclear choices such as the size of the window in which the approximation takes place and the nature of the approximating function. The idea advocated in this and previous work is that it is not necessary to compute \tilde{g} itself, but that it is sufficient to know by which class of continuous functions the digitized function can be approximated well, and to use a difference operator that is known to be optimal for this class. We introduce a criterion that will be used to select an optimal class of approximating functions, and an optimal window size.

We use a continuous real function $\tilde{g} : \mathbb{R}^m \rightarrow \mathbb{R}$ to approximate a digitized function $f : \mathbb{Z}^m \rightarrow \mathbb{Z}$. To approximate the value of a differential at a point x_0 , it is sufficient to approximate f in a finite subset $D \subset \mathbb{Z}^m$ containing x_0 . For all $x \in D$, $f(x) \simeq \tilde{g}(x)$ and $|f - \tilde{g}| < \epsilon$ is written as a shorthand for $|f(x) - \tilde{g}(x)| < \epsilon$.

The shift operator σ^j is defined by $\sigma^j f(x) = f(x+j)$, for $x, j \in \mathbb{Z}^m$. The functional composition of shift operators can be expressed as a multiplication of polynomials, i.e. $\sigma^j \sigma^k f = \sigma^{j+k} f$. A difference operator P can be represented as a polynomial in σ with non-negative, bounded exponents, that is $P = \sum p_j \sigma^j$, and $P \in \mathbb{R}[\sigma]$, the ring of polynomials in σ . We write $P\tilde{g} = 0$ as a shorthand for $\sum p_j \sigma^j \tilde{g}(x) = 0$, $x, j \in \mathbb{Z}^m$. If we write that $|Pf - P\tilde{g}| < \epsilon$, this means that $|Pf(x) - P\tilde{g}(x)| < \epsilon$ for all x for which

$Pf(x)$ is well defined, that is $(x + j) \in D$ for every non-vanishing coefficient p_j of the difference operator P . Since polynomial ideals are usually defined for polynomials with non-negative exponents, we will assume, without loss of generality, that D is a finite rectangular neighborhood containing only points with non-negative coordinates.

An important concept is the ideal $I = \langle P_1, P_2, \dots \rangle$ generated by a set of difference operators P_i . I consists of all operators $P = \sum_i S_i P_i$, where the S_i are arbitrary polynomials in σ . Often the ideal I will be represented by a Groebner basis [13].

Difference operators can be represented by templates. A two-dimensional difference operator $P = \sum p_j \sigma^j = \sum_{j_x, j_y} p_{j_x j_y} \sigma_x^{j_x} \sigma_y^{j_y}$, $j \in \mathbb{Z}^2$ is represented by a two-dimensional template:

$$\begin{array}{|c|c|c|} \hline p_{00} & p_{10} & p_{20} \\ \hline p_{01} & p_{11} & \dots \\ \hline \end{array} \dots \tag{1}$$

We use the convention that the box at the upper left corner corresponds to p_{00} . Boxes with vanishing coefficients are either not drawn, or drawn as empty boxes.

Let L be the differential operator that must be replaced by a difference operator. The goal is to select a class G of fitting functions \tilde{g} , and a difference operator $Q = \sum q_j \sigma^j$ that satisfies $Q\tilde{g} = L\tilde{g}$ for every $\tilde{g} \in G$. If \tilde{g} is an approximation for f such that $|f - \tilde{g}| < \epsilon$ in the domain D , then, because all operators are linear, we have

$$|Qf - L\tilde{g}| < \epsilon \sum |q_j|, \text{ in } D. \tag{2}$$

Hence, the difference operator Q is a good approximation for the differential operator L , provided G contains at least one good approximation \tilde{g} for f . It is important, however, to understand that the total error $\epsilon \sum |q_j|$ is due to two successive approximations. The error ϵ arises when f is replaced by some continuous fitting function \tilde{g} , a step which cannot be avoided if we want to compute differentials. The additional error $\sum |q_j|$ is due to the replacement of the differential operator L by a difference operator Q , which avoids the explicit computation of the continuous approximation. The above technique only works if two distinct requirements are satisfied. The approximation (2) is only valid provided $Q\tilde{g} = L\tilde{g}$ holds for the approximation function. On the other hand the approximation error ϵ should be as small as possible. The best way to satisfy both requirements is to define a class of fitting functions by a set of difference equations $P_i \tilde{g} = 0$. The difference operators P_i span an ideal $I = \langle P_1, P_2, \dots \rangle$, which means that the ideal I can be used to define the class of fitting functions.

Using an ideal of difference operators to define a class of fitting functions has two benefits. First, it becomes easy to verify whether the class of fitting functions contains at least one member \tilde{g} such that $|f - \tilde{g}| \leq \epsilon$. In fact, if $|f - \tilde{g}| \leq \epsilon$, then the inequality $|f - \tilde{g}| \leq \epsilon$ implies $|Pf| \leq \epsilon|P|$ for each $P \in I$, since a difference operator is a linear operator and $P\tilde{g} = 0$. $|P| = \sum |p_j|$ where p_j are the coefficients of P . One can prove that the converse is also true [6], i.e. there is a finite set of difference operators P in I , such that $|f - \tilde{g}| \leq \epsilon$ holds, provided $|Pf| \leq \epsilon|P|$ for all P .

Features can be detected without error by verifying only a finite number of inequalities, when the solution space of the difference equations is a finite linear vector space. Assume that the solution set of the partial difference equations $P_1 g = 0, \dots, P_n g = 0$

can be written as a linear vector space with g_1, \dots, g_l as a basis: $\alpha_1 g_1 + \dots + \alpha_l g_l$. Let K_D be the set of all difference operators P_i of the form

$$\begin{pmatrix} g_1(x_1) & \dots & g_l(x_1) & \sigma^{x_1} \\ \dots & & & \\ g_1(x_{l+1}) & \dots & g_l(x_{l+1}) & \sigma^{x_{l+1}} \end{pmatrix} \tag{3}$$

with the points $x_j \in D$. The operators of K_D are written as determinantal expressions of the coefficients $g_j(x_j)$ and the shift operators σ^{x_j} . Since D is finite, K_D is finite. The operators P in K_D are all operators for which Pf is well defined. When we assume that D is rectangular, there is a unique point x_0 in D with smallest coordinates. Let $\sigma^{-x_0} K_D$ denote the polynomials of K_D multiplied by σ^{-x_0} . The polynomials $\sigma^{-x_0} K_D$ all have non-negative exponents, but some of the exponents will be zero. If D is chosen not too small then the polynomials in $\sigma^{-x_0} K_D$ will generate the entire ideal $I = \langle P_1, \dots, P_n \rangle$. In practice, for two-dimensional operators, it is usually sufficient that D has size at least $(a+2) \times (b+2)$ when the leading monomial of the fitting functions is $x^a y^b$, or smaller. In that case, for f to be in $G_{\epsilon;D}$, it is sufficient to verify $|Pf| \leq \epsilon|P|$ for all P in the finite set K_D [6]. In other words, it is sufficient to verify a finite set of inequalities to determine whether a discrete function f can be approximated well by at least one function in a given function class. Given a domain D , if there is a $\tilde{g} \in G$ such that $|f - \tilde{g}| \leq \epsilon$, we denote this as $f \in G_{\epsilon;D}$.

The second benefit is that it is possible to find an optimal difference operator $R_{G;D}$ for all $f \in G_{\epsilon;D}$, with arbitrary ϵ . Suppose we have at least one operator Q such that $Q\tilde{g} = L\tilde{g}$, then $R\tilde{g} = L\tilde{g}$ for any R for which $R - Q \in I$. Thus R can be chosen from an infinite set of difference operators. The optimization criterion is that $|R|$ should be as small possible.

The function classes used in this paper and their respective Groebner bases are en-listed in Table 1. We shall work with lattices of ideals or, equivalently, lattices of function classes. Suppose we have a set of function classes G_j and a set \mathcal{I} of corresponding

Table 1. Functions and corresponding Groebner bases with lexicographic ordering $\Delta_x > \Delta_y$

$G_1 : \alpha_1$	$\langle \Delta_x, \Delta_y \rangle$
$G_2 : \alpha_1 x + \alpha_2$	$\langle \Delta_x^2, \Delta_y \rangle$
$G_3 : \alpha_1 y + \alpha_2$	$\langle \Delta_y^2, \Delta_x \rangle$
$G_4 : \alpha_1(x + y) + \alpha_2$	$\langle \Delta_x - \Delta_y, \Delta_y^2 \rangle$
$G_5 : \alpha_1 x + \alpha_2 y + \alpha_3$	$\langle \Delta_x^2, \Delta_x \Delta_y, \Delta_y^2 \rangle$
$G_6 : \alpha_1 xy + \alpha_2 x + \alpha_3 y + \alpha_4$	$\langle \Delta_x^2, \Delta_y^2 \rangle$
$G_7 : \alpha_1(x + y)^2 + \alpha_2(x + y) + \alpha_3$	$\langle \Delta_x - \Delta_y, \Delta_y^3 \rangle$
$G_8 : \alpha_1(x + y)^2 + \alpha_2 x + \alpha_3 y + \alpha_4$	$\langle \Delta_x^2 - \Delta_y^2, \Delta_y(\Delta_x - \Delta_y), \Delta_y^3 \rangle$
$G_9 : \alpha_1(x^2 + y^2) + \alpha_2 x + \alpha_3 y + \alpha_4$	$\langle \Delta_x^2 - \Delta_y^2, \Delta_x \Delta_y, \Delta_y^3 \rangle$
$G_{10} : \alpha_1 x^2 + \alpha_2 y^2 + \alpha_3 x + \alpha_4 y + \alpha_5$	$\langle \Delta_x^3, \Delta_x \Delta_y, \Delta_y^3 \rangle$
$G_{11} : \alpha_1(x^2 + y^2) + \alpha_2 xy + \alpha_3 x + \alpha_4 y + \alpha_5$	$\langle \Delta_x^2 - \Delta_y^2, \Delta_x \Delta_y^2, \Delta_y^3 \rangle$
$G_{12} : \alpha_1 x^2 + \alpha_2 y^2 + \alpha_3 xy + \alpha_4 x + \alpha_5 y + \alpha_6$	$\langle \Delta_x^3, \Delta_x^2 \Delta_y, \Delta_x \Delta_y^2, \Delta_y^3 \rangle$
$G_{13} : \alpha_1 x^3 + \alpha_2 x^2 y + \dots + \alpha_n$	$\langle \Delta_x^4, \Delta_x^3 \Delta_y, \Delta_x^2 \Delta_y^2, \Delta_x \Delta_y^3, \Delta_y^4 \rangle$
$G_{14} : \alpha_1 x^4 + \alpha_2 x^3 y + \dots + \alpha_n$	$\langle \Delta_x^5, \Delta_x^4 \Delta_y, \Delta_x^3 \Delta_y^2, \Delta_x^2 \Delta_y^3, \Delta_x \Delta_y^4, \Delta_y^5 \rangle$

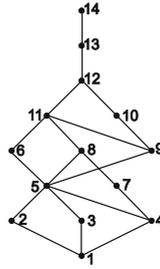


Fig. 1. Lattice of the operator ideals shown in Table 1

ideals I_j . Then \mathcal{I} is a lattice provided for each pair of ideals I_k, I_l in \mathcal{I} the intersection (or meet) of I_k and I_l , as well as their union (or join) are also included in \mathcal{I} . The ideals of Table 1 form in fact the lattice shown in the Figure 1.

3 Computing Optimal Operators

We will now go into more detail with regard to the optimization process for the optimal difference operator $R_{G;D}$, i.e. the operator for which $|R|$ is as small as possible. First, we give a general procedure for finding all operators with a predefined shape for a given function class. Then we can compute the optimal operator for each class by solving the optimization problem $\min(|R|)$.

3.1 Shaped Operators

Difference operators S with a predefined shape have the advantage that known properties of the operator can be taken into account so that the optimization algorithm is sped up. For example, if it is known that there exists a optimal symmetric operator, we can use this fact by starting with a predefined symmetrical shape, which reduces the optimization space considerably.

A general form for all operators of a predefined shape can be derived as follows. First compute a reduced operator modulo the function class for both the difference operator Q , with $Q\tilde{g} = L\tilde{g}$, and the operator S with the predefined shape. Set the reduced predefined operator equal to the reduced shaped operator and solve the resulting system of linear equations. When this system has no solution, there is no operator with the predefined form. When a solution is found, some variables are eliminated to obtain an expression in which the number of remaining variables equals the number of dimensions of the solution space. The resulting operator represents symbolically all correct operators of the predefined shape.

Example. We illustrate this procedure for the function class G_7 , and the Laplacian $L = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ as differential operator. For functions in G_7 , the Laplacian L can be replaced by the difference operator $Q = 2\Delta_y^2 = 2\sigma_y^2 - 4\sigma_y + 2$. The difference operator Q is shifted such that the Laplacian is computed in the center of a 5x5 square:

$Q' = (2\sigma_y^2 - 4\sigma_y + 2)\sigma_x^2\sigma_y^2$. When Q' is reduced modulo the ideal $I_{G_7} = \langle (\sigma_x - 1) - (\sigma_y - 1), (\sigma_y - 1)^3 \rangle$, we obtain $Q' \bmod I = 2 - 4\sigma_y + 2\sigma_y^2$.

Note that the reduced operators $Q' \bmod I$ are in general not symmetric in σ_x, σ_y , because the Groebner basis of I is computed according to a predefined lexicographic ordering relation between the variables, $\sigma_x > \sigma_y$. Since the same Groebner basis is used to reduce the operator S , this will not affect the final result. We choose for example a 5x5 difference operator S of the form

$$\begin{matrix}
 a & b & c & b & -a \\
 b & d & e & -d & b \\
 c & e & f & e & c \\
 b & -d & e & d & b \\
 -a & b & c & b & a
 \end{matrix} \tag{4}$$

with coefficient $b = 0$. The reduction of the operator S yields

$$\begin{aligned}
 S \bmod I &= (16a + 20c + 4d + 14e + 3f) \\
 &+ (-32a - 48c - 8d - 36e - 8f)\sigma_y \\
 &+ (16a + 32c + 4d + 26e + 6f)\sigma_y^2.
 \end{aligned} \tag{5}$$

We obtain 3 linear equations by setting $Q' \bmod I = S \bmod I$, which enables us to eliminate 3 variables. The result is a general expression for all correct operators of the shape (4) in which the variables a, c, d can be chosen freely, and $b = 0, e = 1 - 8a - 4c - 2d, f = -4 + 32a + 12c + 8d$.

Several function classes will yield operators with the same form. For example, the classes $G_7, G_8, G_9, G_{10}, G_{11}, G_{12}$ all have the same symmetrical operator. This means that these function classes will also have the same optimal operator of that shape.

3.2 The Optimal Operator

An operator is optimal when the sum of the absolute values of its coefficients is minimal. So once the general form of an operator is known, an optimal operator can be found by solving a large set of linear programming problems. The cost of the operator in the previous example is $4|a| + 4|c| + 4|d| + 4|1 - 8a - 4c - 2d| + |-4 + 32a + 12c + 8d|$. Although this expression is non-linear, the optimum can be found by solving 2^5 small linear programming problems. In each problem, we make an assumption for each absolute value about the sign of the term and replace it by a linear term. Since there are 5 terms in this example, there are 2^5 possible combinations. For our example, we obtain an optimal operator with cost 1/2 and the template shown in Fig. 2(b).

Several function classes may have the same optimal operator. Figure 2 shows the optimal operators for some function classes and different window sizes. The costs for all functions classes and windows is given by Table 2.

In Table 2, the set of linear function classes comprises G_1, G_2, G_3, G_4 and G_5 ; the quadratic classes are G_9, G_{10}, G_{11} and G_{12} ; the cubic class is G_{13} ; the class of fourth order is G_{14} . Although G_6 is not a class of linear functions, with respect to the Laplacian it behaves as a linear function. We use the term quadratic symmetric to denote the set

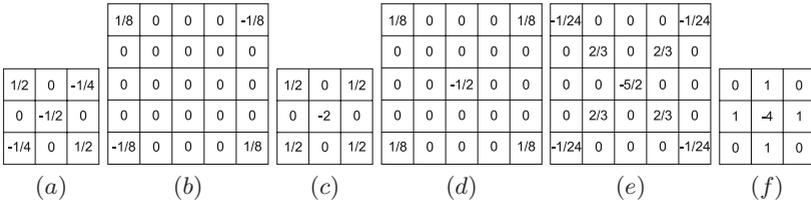


Fig. 2. Optimal Laplacian operator templates for several sets of function classes and different window sizes: $\{G_7, G_8\}(3 \times 3)$ (a), $\{G_7, G_8\}(5 \times 5)$ (b), $\{G_9, G_{10}, G_{11}, G_{12}\}(3 \times 3)$ (c), $\{G_9, G_{10}, G_{11}, G_{12}, G_{13}\}(5 \times 5)$ (d), G_{14} (e), widely used template (f)

Table 2. Cost of optimal operators for different sets of function classes and window sizes

\ functions window	linear	quadratic symmetric	quadratic	cubic	4th order
3x3	0 (1)	2	4	n.e.	n.e.
5x5	0 (1/4)	1/2	1	1	16/3
7x7	0 (1/9)	2/9	4/9	4/9	9/5

of function classes $\{G_7, G_8\}$, which are symmetric in the second degree terms. There are no 3×3 operators that correctly compute the Laplacian for cubics and higher order functions.

For a given window size, the cost increases when the function class becomes more general. In fact, Table 1 shows that $I_{12} \subset I_{10}$, i.e. G_{12} contains G_{10} as a subset, and the functions in G_{10} have to satisfy the difference equations from I_{12} plus some extra difference equations. As a result, the class of operators from which an optimal operator has to be chosen is larger for I_{10} than for I_{12} . Second, for a given function class the cost decreases when the window size increases. In fact, the operators for a larger window contain the operators for a small window as a special case.

For linear functions, the optimal operator is the zero operator with cost equal to zero. This is correct since the Laplacian of a linear function (and of G_6) is zero, a value which can be estimated without error. This may give rise to some anomalies when we have to select an appropriate function class and window size. In the next section we discuss how these anomalies can be avoided by introducing an artificial cost for linear functions (shown between brackets in Table 2).

4 Optimal Operator Selection

We have now assigned an optimal operator to each function class for each window size. We now propose a selection criterion for the best operator with regard to the operator cost and the local image properties, i.e. the approximation cost for a certain function class and window size. Our selection method is then validated by experiments on artificial images of digitized surfaces.

4.1 How to Select Function Class and Window Size?

We noticed earlier that the function classes form a poset with subset inclusion as ordering relation, and that the same difference operator is optimal for a subset of function classes. Therefore the number of classes to be selected from can be limited. When two classes have the same optimal operator for a given window size, we consider only the most general class because it yields the smallest approximation error ϵ . ϵ is a measure for the local fit of the approximating continuous function to the digitized surface. A feature detection template P_i yields a cost ϵ_i when applied to the image part f :

$$\epsilon_i = \frac{|P_i f|}{|P_i|} \quad (6)$$

with $P_i \in K_D$ (3) for that function class. The fitting cost ϵ is then obtained as $\epsilon = \max_i(\epsilon_i)$ for a limited subset of feature detectors P_i (usually 10-15 suffices).

For practical reasons, we only consider windows of either 3x3 or 5x5 in our experiments. The following classes are available in our experiments:

- The function classes G_1, G_2, G_3, G_4, G_5 and G_6 form a subset of functions for which the optimal difference operator is the zero operator, for both the 3x3 and 5x5 window size. G_6 is the most general of these functions, and is therefore used as a representative for this set of classes.
- G_8 represents the set of quadratic function classes $\{G_7, G_8\}$. The optimal operators are shown in Fig. 2 for the 3x3 and 5x5 templates.
- Quadratic functions as in G_9, G_{10}, G_{11} and G_{12} are represented by G_{12} , with the operator shown in Fig. 2 as the optimal operator for the 3x3 templates. For 5x5 templates, the optimal operator generated for these classes is the same as that for class G_{13} . Therefore, we use G_{13} to generate the feature detection templates for this subset of function classes.

Up to now, we have an optimal operator for a given function class and a given window. When the window size increases, the cost goes down for a given function class. When the window size is fixed, the cost goes down when the function class becomes less general. Therefore, on the one hand it makes sense to use a large window with a more specialized function class, on the other hand this will increase the fitting cost. We now propose a criterion which seeks an optimum between these two choices: use the function class and window size for which $\epsilon|R|$ is minimal.

The product $\epsilon|R|$ represents the fitting cost ϵ as well as the operator cost $|R|$ so that the image surface must locally be well approximated by the function class and the error introduced by using a difference operator must be small. Unfortunately, this criterion does not work for zero operators as their operator cost $|R|$ is always zero, and therefore also the estimation error $\epsilon|R|$ is zero. Theoretically, this is correct because $\epsilon|R|$ represents the error with which we can estimate the value of $L\tilde{g}$. Since the Laplacian of a linear function is always zero, we can in fact estimate the value of the Laplacian without error. The fitting cost does not count in this case. One solution would be to define a more elaborate mixed cost $h(\epsilon, \epsilon|R|)$. We propose a simpler solution by introducing a non-zero cost for zero operators. By observing Table 2, one sees that the cost $|R|$ often goes down by a factor 2 when the function class gets more specialized. For example,

in a 3×3 window, the cost is 4 for quadratic functions, and 2 for quadratic symmetric functions. We shall extrapolate this trend and use a cost for linear functions which is half the cost of symmetric quadratic functions, as shown between brackets in Table 2. Thus, quadratic or higher order functions will only be used if their fitting cost is at least less than one half the fitting cost of linear functions. By a fortunate coincidence, the cost used in a 3×3 window for linear functions is simply the fitting cost ϵ itself.

4.2 Evaluation

To verify whether this selection method yields good results, first the following question must be answered: Are the correct fitting classes chosen, i.e. is $\epsilon|R|$ locally minimal for the correct function class? Experimental data is obtained for images of digitized surfaces as shown in Fig. 3(a). The surfaces are composed by functions of different classes so that the exact value of the Laplacian is known beforehand for each pixel in the image. This allows for a comparison of the values for the Laplacian obtained by the proposed method, the hierarchical method discussed in [7] and the widely used operator as shown in Fig. 2(f).

We reconsider the hierarchical method [7], which determines the function class by descending a tree-structure in which three consecutive nodes respectively represent G_5 , G_8 and G_{12} . Feature detection is performed by evaluating

$$\frac{|P_i f|}{|P_i|} \leq \epsilon_{th} \quad (7)$$

with ϵ_{th} a predefined threshold for a limited number of feature detection templates P_i (3). If this test succeeds at a node in the tree, the corresponding optimal difference operator is chosen. Although this method shows better results than the application of the classical Laplacian template, this approach still has some drawbacks. A first problem is the extension of the tree for templates of different sizes. Which size should be preferred in the hierarchical composition of the tree? Second, why should the first node where the test succeeds always be preferred? The second or third class could yield a better approximation for the digitized image surface. If we select an appropriate Laplacian operator by descending the tree, we notice that the zero-operator is often chosen, even for quadratic functions. In those cases, a linear function locally is a good enough approximation for the digitized image. Then the zero operator is applied, which does not correspond to the actual Laplacian value. A related problem is the choice of the threshold ϵ_{th} .

In the now proposed approach, the appropriate function class is chosen by the criterion $\min(\epsilon|R|)$ over all function classes. $\epsilon|R|$ is easily computed for different template sizes and discriminates well between different classes. Mostly the correct function class is chosen for each image part throughout the experiments. An example is shown in Fig. 3: the function class for which $\epsilon|R|$ is minimal is shown for each pixel. Fig. 3 shows that the use of larger feature templates proves its worth. A larger image region then influences the outcome of the Laplacian operator so that its result is more reliable. We notice however that in some cases the 'incorrect' class is chosen, e.g. the linear case is chosen for quadratic cases, when the image surface is locally represented by a rather flat

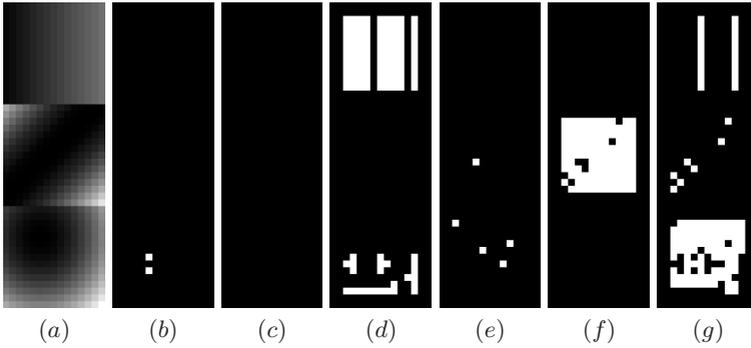


Fig. 3. Figure (a) shows a digitized image composed of three digitized surfaces: linear, quadratic symmetric and quadratic (from top to bottom). The next images respectively represent the function classes $G_6(3 \times 3)$ (b), $G_8(3 \times 3)$ (c), $G_{12}(3 \times 3)$ (d), $G_6(5 \times 5)$ (e), $G_8(5 \times 5)$ (f) and $G_{13}(5 \times 5)$ (g). When the criterion $\epsilon|R|$ is minimal for that function class and window size in a pixel, that pixel is indicated as white.

quadratic. Notice that only a limited subset of feature detectors P_i is used to compute ϵ . We must also notice that we give preference to larger windows and functions of higher order when $\epsilon|R|$ is minimal for multiple classes. This is e.g. the case for most pixels in the upper part of the image in Fig. 3, where $\epsilon|R| = 0$ for all 3×3 templates, so that the class with polynomials of the highest order (G_{12} in this case) is chosen.

4.3 Computation of the Difference Operator

It is not sufficient to select the correct function class. Does the optimal operator for a specific function class also yield a good estimation for the exact value of the difference operator? Is the estimated value considerably better than the value obtained by applying the widely used template for the difference operator?

When we compute the Laplacian for image surfaces composed of known digitized functions with the optimal operator for that function class, we can compare that result to the actual Laplacian value for the continuous functions. An example is shown in Fig. 4. We notice that the operators with a 5×5 template estimate the actual value more accurate than the value obtained by the corresponding 3×3 operator template. This is expected as the maximum error is $|R|$, which is smaller for larger templates. The experiments also show that in most cases the Laplacian is estimated much better than the expected maximum error of that operator. The mean and maximum error on the estimation of the actual value are respectively up to 8 and 2 times smaller than $|R|$ for the optimal difference operator when the class is known. The estimation is in almost all cases better than that of the default operator. The mean error on our estimation is much smaller (up to more than 10 times) than the mean error on the default kernel.

When we combine the class selection method and the application of the optimal difference operator for the selected class, we can evaluate the results of the proposed

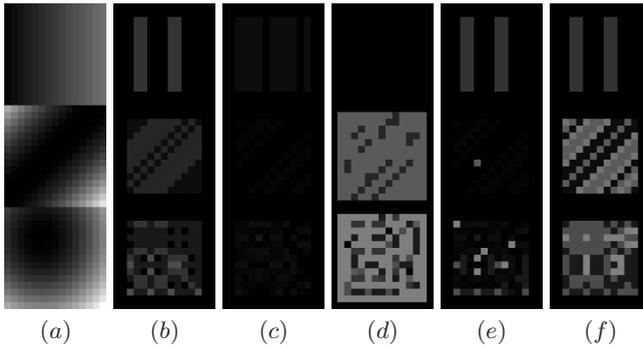


Fig. 4. An image (a) composed of three digitized surfaces: linear, quadratic symmetric and quadratic (from top to bottom). The other images indicate the absolute error on the estimation of the Laplacian in gray values, with black to white corresponding to the range $[0,5]$. In (b) the Laplacian is computed in each of the three image parts with respectively the optimal 3×3 operator for class G_6 , G_8 and G_{12} . In (c) the Laplacian is computed with the optimal 5×5 operator for G_6 , G_8 and G_{13} in the corresponding image parts. The error on the Laplacian is indicated when the operator is selected respectively by the hierarchical tree (d), by the criterion $\min(\epsilon|R|)$ (e) and for the default Laplacian kernel (f).

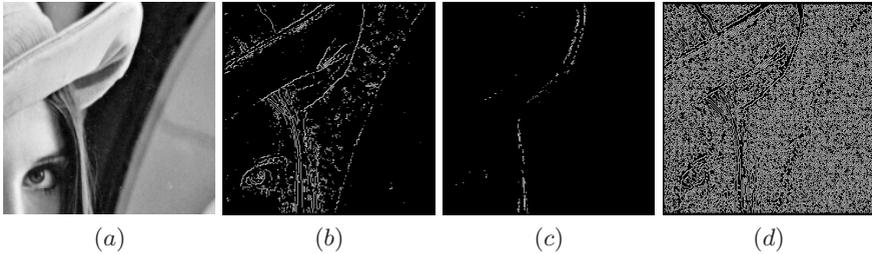


Fig. 5. Original image (a), edges detected as zero crossings after Laplacian computation on the unsmoothed image by the proposed method (b), by the hierarchical method (c) and by the default method (d)

method. When compared to the results of the hierarchical tree, we notice a considerable improvement, as can be observed for the example in Fig. 4. We notice that the zero operator is frequently chosen not only for linear, but also for quadratic surfaces as it appears first in the tree. This problem is solved by the proposed method. As already noted, the application of optimal operators of larger sizes brings increased accuracy. Even more, the experiments show that these cases are preferred over the less accurate smaller templates. It is also proven by the experiments that the proposed method always outperforms the default method. Both the mean and the maximum error of the estimated value for the Laplacian compared to the actual value is always greater in case of the default estimation method. The mean error is in the order of a few tenths for our method, while it attains mean error values greater than one for the default method.

5 Concluding Remarks

We present a procedure to replace a differential operator by a difference operator optimal for a specific class of functions. The criterion $\min(\epsilon|R|)$ determines which operator to use for a specific image part. Experiments on digitized functions show that the Laplacian is indeed better estimated by our method than by the widely used Laplacian kernel. A better estimation of the Laplacian offers a considerable advantage in feature detection applications, for example edge detection by looking for zero crossings of the Laplacian. A better estimate of the Laplacian results in a more accurate computation of the zero crossings. In particular, we expect a considerable improvement for large noisy image regions with homogenous gray values. Regions around edges can be well approximated by the quadratic and higher order function classes, so that the error on the computation of the Laplacian will decrease when our criterion is applied to select the best operator. In Fig. 5, we apply our method to compute the Laplacian for the unsmoothed Lena image. When we compare the edges obtained from zero crossings of the Laplacian, we notice that the proposed method gives better results than the hierarchical method, and much better than the widely applied method for the discrete Laplacian.

References

1. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2), 91–110 (2004)
2. Lindeberg, T.: Discrete Derivative Approximations with Scale-Space Properties: A Basis for Low-Level Feature Extraction. *J. of Mathematical Imaging and Vision* 3, 349–376 (1993)
3. Lachaud, J.O., Vialard, A., de Vieilleville, F.: Analysis and Comparative Evaluation of Discrete Tangent Estimators. In: Andrès, É., Damiand, G., Lienhardt, P. (eds.) DGCI 2005. LNCS, vol. 3429, pp. 240–251. Springer, Heidelberg (2005)
4. Gunn, S.: On the discrete representation of the Laplacian of Gaussian. *Pattern Recognition* 32, 1463–1472 (1999)
5. Demigny, D., Kamlé, T.: A Discrete Expression of Canny’s Criteria for Step Edge Detector Performances Evaluation. *IEEE Trans. Patt. Anal. Mach. Intell.* 19, 1199–1211 (1997)
6. Veelaert, P.: Local feature detection for digital surfaces. In: *Proceedings of the SPIE Conference on Vision geometry V*, SPIE, vol. 2826, pp. 34–45 (1996)
7. Teelen, K., Veelaert, P.: Improving Difference Operators by Local Feature Detection. In: Kuba, A., Nyúl, L.G., Palágyi, K. (eds.) DGCI 2006. LNCS, vol. 4245, pp. 391–402. Springer, Heidelberg (2006)
8. Veelaert, P., Teelen, K.: Feature controlled adaptive difference operators. *Discrete Applied Mathematics* (preprint, submitted, 2007)
9. Oberst, U.: Multidimensional constant linear systems. *Acta Appl. Math.* 20, 1–175 (1990)
10. Oberst, U., Pauer, F.: The Constructive Solution of Linear Systems of Partial Difference and Differential Equations with Constant Coefficients. *Multidim. Systems and Signal Processing* 12, 253–308 (2001)
11. Gerdt, V., Blinkov, Y., Mozhilkin, V.: Groebner Bases and Generation of Difference Schemes for Partial Differential Equations. *Symmetry, Integrability and Geometry: Methods and Applications* 2 (2006) (Paper 051, arXiv:math.RA/0605334)
12. Stoer, J., Witzgall, C.: *Convexity and Optimization in Finite Dimensions I*. Springer, Berlin (1970)
13. Cox, D., Little, J., O’Shea, D.: *Ideals, Varieties and Algorithms: an Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, New York (1992)