A Multigrid Approach to Image Processing

Paul M. de Zeeuw

CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

Abstract. A second order partial differential operator is applied to an image function. By using a multigrid operator known from the so-called approximation property, we derive a new type of multiresolution decomposition of the image. As an example, the Poisson case is treated in-depth. Using the new transform we devise an algorithm for image fusion. The actual recombination is performed on the image functions on which the partial differential operator has been applied first. A fusion example is elaborated upon. Other applications can be envisaged as well.

Keywords: This work was carried out under project CWI - PNA4.2 "Image Representation and Analysis".

1 Introduction

We seek to integrate multigrid methods [4] for the numerical solution of partial differential equations (PDEs) with image processing methods. Modeling by PDEs emerges as a powerful approach to the formulation of image processing problems. An example is the *level set method* [12] originating from computational physics which was transferred to image analysis [15] in the mid 90s. It found important applications like restoration of degraded images and image segmentation.

There exists a repository of modern methods in numerical mathematics from which image processing can benefit [2, 22]. In particular we allude to multigrid methods for the solution of PDEs, hereby involving a multiresolution approach. This method, which exists for a few decades, accelerates a basic iterative technique by means of coarse grid corrections, resolving the low-frequent components on coarser grids with increasing mesh-size (see Figure 2). If well-designed, this method holds out the prospect of optimal computational complexity. It has found applications in the computationally highly demanding computational fluid dynamics. One observes that in a parallel development, multiresolution has become an important ingredient for image processing as well.

We devise and investigate a new image processing method which involves the concepts of image transforms, PDEs and multiresolution all in one. Instead of the more traditional multiresolution transforms, we propose to transform by means of discretized partial differential operators on a sequence of increasingly coarsened grids.

Terzopoulos [19] was the first to apply multigrid for image analysis. More recently, the use of multigrid for image processing purposes has been proposed

R. Kimmel, N. Sochen, J. Weickert (Eds.): Scale-Space 2005, LNCS 3459, pp. 396–407, 2005.

by Acton [1], Kimmel et al. [9], Shapira [16] and others. However, its use was restricted to the efficient solution of partial differential equations (typically diffusion and Euler-Lagrange equations) which could also be achieved by other means.

In this paper multigrid operators are used as an intrinsic and indissoluble part of the transform. Here and now the transform is applied to image fusion but it may have future implications for image segmentation and edge detection. The paper is organized as follows. After preliminaries in Section 2 we are ready for the introduction of the *multigrid image transform* in Section 3 and the *multigrid fusion algorithm* in Section 4. We end up with concluding remarks.

2 Recapitulation and Preliminaries

Image fusion seeks to combine images in such a way that all the salient information is put together into (usually) one image suitable for human perception or further processing. It is hard to overrate the practical importance of image fusion. For example, for the purpose of surveillance one and the same scene is recorded by cameras operating for different bands of light and needs to be displayed onto one screen, preferably in real-time. Similar applications exist in the fields of defense, geoscience, robotics and medical imaging.

The multigrid method solves discretized elliptic, parabolic and hyperbolic PDEs as well as integral equations by accelerating a basic iterative solution process through adequate coarse grid corrections. If well designed and implemented, multigrid algorithms offer the possibility of computational complexity and storage which are linearly proportional to the number of grid-points. For a historical overview of the development see Wesseling [23]. Today, it continues to evolve from an advanced numerical technique towards an established method. Nowadays extensive literature is available on multigrid. Here we merely point to Brandt [4], Hackbusch [8], Wesseling [23] and (more recent) to Trottenberg et al. [21] and Shapira [16].

Firstly, we discuss the multiresolution approach to image fusion. Secondly, we briefly discuss multiresolution transforms. Thirdly, we recapitulate on multigrid.

2.1 Multiresolution Image Fusion

There exist various categories of techniques for image fusion, but we merely consider methods by means of the multiresolution (MR) approach. The basic idea is demonstrated by Figure 1 (cf. [14–Figure 6.6]). At the decomposition stage the input images (i_A, i_B, i_C, …) are transformed into multiresolution representations ($m_A, m_B, m_C, \dots$). The transform is symbolized by Ψ. At the combination stage ($\mathcal{C}$) the transformed data are fused. In the context of wavelets, Li et al. [10] proposed to apply the *maximum selection* rule for the detail coefficients as fusion rule. For instance, in the case of three input images, we select from each triplet of geometrically corresponding detail coefficients the one that is largest in absolute value. From the composite multiresolution representation m_F thus obtained, the fused image i_F is derived by application of the backtransform Ψ^{-1}. Many far

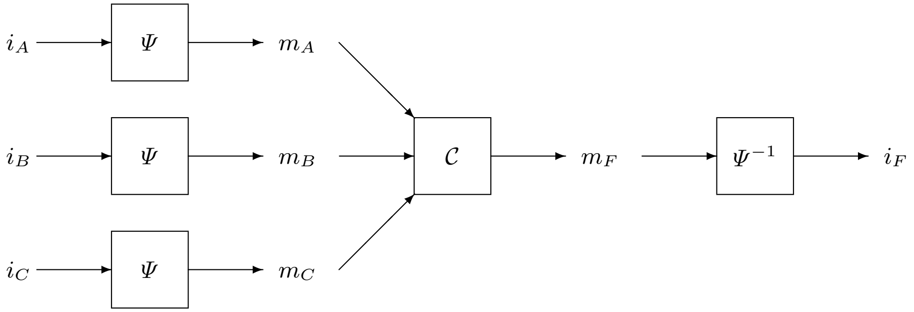

Fig. 1. MR image fusion scheme. Left: MR transform Ψ of the sources; middle: combination in the transform domain; right: inverse MR transform Ψ^{-1} of the composite representation

more sophisticated fusion rules have been invented by now, e.g. one that is based on maximizing luminance contrast [20]. For an overview see Piella [13, 14].

2.2 Transform and Backtransform

What schemes like Laplacian pyramids [5], gradient pyramids [6], steerable pyramids [17], wavelets [11]), and the lifting scheme [18] have in common is that they involve filters for the decomposition and the reconstruction, down- and upsampling towards and from scales and storage of approximation coefficients and detail coefficients collected in so-called bands.

Part of the new transform that we propose here involves the discretized version of $-\nabla \cdot (D\nabla u)$ (where $D(x, y)$ is a positive definite 2×2 matrix function, for the time being assumed to be a constant times the identity matrix) that is applied to the image. One observes that hereby the outcome vanishes at smooth regions of an image but becomes substantial where edges occur. The transforms are applied with respect to a sequence of increasingly coarsened grids, see Figure 2. At a certain stage the (back)transform involves the solution of large linear systems of equations as it needs to invert the said discrete operators again. However, the costs of solution of such systems need not to be prohibitive anymore, e.g. see [3]. The procedure is explained in much detail in Section 3 after a recapitulation of a particular multigrid algorithm.

2.3 Multigrid Algorithm

De Zeeuw (this author) published a paper on a robust multigrid algorithm for the numerical solution of diffusion and convection-diffusion problems [24]. The algorithm has been implemented and exists by the name of MGD9V. This paper is here of particular importance and we recapitulate particular items that we need. For the multigrid method to be discussed we consider a set of increasingly coarser grids (vertex-centered):

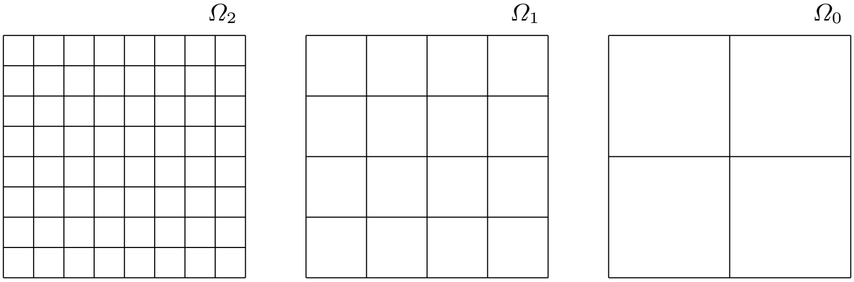

Fig. 2. Example sequence of increasingly coarsened grids used in multigrid (vertex-centered)

$$\Omega_n \supset \Omega_{n-1} \supset \ldots \Omega_k \supset \ldots \supset \Omega_0.$$

The grids are described as follows:

$$\Omega_k \equiv \{(x_i, y_i) \mid x_i = o_1 + (i-1)h_k, y_i = o_2 + (j-1)h_k\} \tag{1}$$

where (o_1, o_2) is the origin and $h_{k-1} = 2h_k$. See Figure 2 for an example. $S(\Omega_k)$ denotes the linear space of real-valued functions on Ω_k

$$S(\Omega_k) = \{g_{\mathbf{k}} \mid g_{\mathbf{k}} : \Omega_{\mathbf{k}} \to \mathbb{R}\},$$

where $g_k \in S(\Omega_k)$ is called a *grid-function.* The algorithm is intended for the solution of linear systems. Its scope is the solution of linear systems resulting from the 9-point discretization of the following general linear second-order elliptic partial differential equation in two dimensions:

$$Lu \equiv -\nabla \cdot (D(x)\nabla u(x)) + b(x) \cdot \nabla u(x) + c(x)u(x) = f(x) \tag{2}$$

on a bounded domain $\Omega \subset \mathbb{R}^2$ with suitable boundary conditions. $D(x)$ is a positive definite 2×2 matrix function and $c(x) \geq 0$. We suppose that Ω is a rectangular domain. It is assumed that the discretization of (2) is performed by a finite element or finite volume technique, leading to

$$L_n \overline{u}_n = f_n \tag{3}$$

where $\overline{u}_n$ and f_n are grid-functions defined on the grid Ω_n. The discretization on the finest grid Ω_n evokes the linear system (3). The grids need to be neither uniform nor rectangular, problem (3) may be discretized on a curvilinear grid.

The code performs only for the scalar case and within the constraints of a regular domain and a structured grid. Incomplete line LU-factorization is used as basic iterative method. Like for other basic iterative methods, the convergence is slow for low-frequent components in the residual. It is accelerated by coarse grid corrections, resolving the low-frequent components on coarser grids with increasing mesh-size. The algorithm of MGD9V is therefore an example of a multigrid method. Let u_n be an approximation of $\overline{u}_n$, the coarse grid correction (CGC) then reads:

$$r_{n-1} = R_{n-1}(f_n - L_n u_n); \tag{4}$$

$$\text{solve } L_{n-1}e_{n-1} = r_{n-1}; \tag{5}$$

$$\tilde{u}_n = u_n + P_n e_{n-1}. \tag{6}$$

Where

$$R_{k-1} \quad : \quad S(\Omega_k) \to S(\Omega_{k-1}), \quad k = n, \ldots, 1 \tag{7}$$

is the restriction operator that transfers the residual from the grid Ω_k onto the coarser grid Ω_{k-1}, and

$$P_k \quad : \quad S(\Omega_{k-1}) \to S(\Omega_k), \quad k = n, \ldots, 1 \tag{8}$$

is the prolongation operator that interpolates and transfers a correction for the solution from the coarser towards the finer grid. The operator L_{k-1} is defined by the sequence of operations

$$L_{k-1} \equiv R_{k-1}L_kP_k, \quad k = n, \ldots, 1. \tag{9}$$

known as the *Galerkin coarse grid approximation.* The diagram of Figure 3 illustrates the coherence of the above mentioned operators. We choose the restriction to be the transpose of the prolongation

$$R_{k-1} = P_k^T, \quad k = n, \ldots, 1. \tag{10}$$

Hence, once P_k has been chosen, R_{k-1} and L_{k-1} follow automatically. The code actually computes the coarse grid matrix of L_{k-1}. Note that by (10) the possible (anti)symmetry of L_k is maintained on the coarser grid. Further, it has been proved [24] that when L_k is a conservative discretization of L and P_k interpolates a constant function exactly, then the Galerkin approximation L_{k-1} is conservative as well. In the case of e.g. the Poisson equation and discretization by bilinear finite elements, bilinear interpolation is the natural choice for P_k. In the case of discontinuous diffusion coefficients a far more sophisticated choice is required [24].

The importance of the CGC can be seen as follows (for pointers to a more rigorous analysis see the earlier listed references). For the sake of argument suppose that the system of stage (5) has been solved exactly. By (9) it follows that after such an ideal coarse grid correction the restriction of the residual vanishes

$$R_{k-1}(f_k - L_k\tilde{u}_k) = 0_{k-1}. \tag{11}$$

This means that at each coarse grid point a weighted average (with non-negative weights) of the fine-grid residual is zero, which implies that the residual consists of short wavelength components only. Such components can be reduced efficiently by a subsequent smoothing (relaxation) step. In practice, the system of stage (5) is not solved exactly. Instead, the algorithm is applied in a recursive manner with respect to the solution of (5). This completes one so-called multigrid cycle.

$$\begin{tikzcd}
S(\Omega_n) \arrow[r, "L_n"] & S(\Omega_n) \arrow[d, "R_{n-1}"] \\
S(\Omega_{n-1}) \arrow[u, "P_n"] \arrow[r, "L_{n-1}"] & S(\Omega_{n-1}) \arrow[d, "R_{n-2}"] \\
S(\Omega_{n-2}) \arrow[u, "P_{n-1}"] \arrow[r, "L_{n-2}"] & S(\Omega_{n-2}) \\
\vdots & \vdots \\
S(\Omega_0) \arrow[r, "L_0"] & S(\Omega_0)
\end{tikzcd}$$

Fig. 3. Diagram of Galerkin approximation

In general, the multigrid method holds out the prospect of a computational complexity which is directly proportional to the number of unknowns. The algorithm of MGD9V comes up to these expectations.

3 The Multigrid Image Transform

We introduce the *multigrid image transform* and discuss some of its properties.

3.1 Definition and Properties

We define the multigrid approximation operator: $E_k : S(\Omega_k) \to S(\Omega_k)$ as follows:

$$E_k \equiv L_k^{-1} - P_k L_{k-1}^{-1} R_{k-1}, \quad k = 1, \ldots, n. \tag{12}$$

This operator plays an important role in convergence proofs in multigrid theory. It is associated with the so-called *approximation property*. Under a certain regularity of the boundary value problem (2), a discretization (3) by (bilinear) finite elements, and P_k is bilinear interpolation, it can be shown that (see Hackbusch [8–§6.3]):

$$\|E_k\|_2 \quad \leq \quad Ch_k^2 \tag{13}$$

where h_k is the mesh-size of Ω_k and $\|\cdot\|_2$ is the Euclidean norm on $S(\Omega_k)$.

Let u_n be an image, defined as a grid-function on $S(\Omega_n)$. Then compute grid-function $f_n = L_n u_n$, for the definition of L_n see (2) and (3). An important example for L is the Poisson operator, this is discussed in Section 4.2. Let

$$f_k \equiv R_k f_{k+1}, \quad k = n-1, \ldots, 0 \tag{14}$$

then we define the *multigrid image transform* or *multigrid image decomposition* as follows

$$\begin{cases} a_0 = L_0^{-1} f_0, \\ d_k = E_k f_k, \quad k = 1, \ldots, n. \end{cases} \tag{15}$$

The a_k are called *approximations* and the d_k are called *details.* The reconstruction counterpart reads:

$$a_k = P_k a_{k-1} + d_k, \quad k = 1, \ldots, n. \tag{16}$$

Proposition 1. *Regarding (3), (7)–(9), (12), (14)–(16) it follows that*

$$L_k a_k = f_k, \quad k = 0, \ldots, n.$$

Proof. By definition, the statement holds for $k = 0$. From decomposition (15) it follows that

$$L_k d_k = L_k E_k f_k = (I_k - L_k P_k L_{k-1}^{-1} R_{k-1}) f_k, \quad k = 1, \ldots, n$$

where I_k is the identity operator on $S(\Omega_k)$. Then multiplying (16) by L_k leads to

$$L_k a_k = L_k P_k a_{k-1} + (I_k - L_k P_k L_{k-1}^{-1} R_{k-1}) f_k = f_k - L_k P_k (a_{k-1} - L_{k-1}^{-1} R_{k-1} f_k).$$

But then, through induction, the proof can be completed at once. □

Hence, the reconstruction (16) with respect to the decomposition (15) is a perfect one.

4 The Multigrid Fusion Algorithm

Firstly we describe fusion algorithms by means of the above transform. Secondly we address the important topic of boundary conditions. Thirdly we elaborate on an example case using the Poisson operator.

We assume to have a set of m multiple input images $\{i_{1,n}, \ldots, i_{m,n}\} \in S(\Omega_n)$ that need to be fused. The decomposition (15)–(16) suggests several options for image fusion. The most basic one is to select from each set of m geometrically corresponding details on each level k the one that is largest in absolute value. This line of research is not pursued in this paper. Instead, we opt here for fusion in the space of right-hand side grid-functions. For that we proceed as follows. Firstly, we compute $f_{j,n} = L_n i_{j,n}$ for $1 \leq j \leq m$. Secondly, we compute $f_{j,k} = R_k f_{j,k+1}$ for $1 \leq j \leq m$ and $k = n-1, \ldots, 0$. At each level k we apply a recombination $\mathcal{C}_k : S(\Omega_k) \times \ldots \times S(\Omega_k \rightarrow S(\Omega_k)$ on $f_{j,k}$:

$$f_k = \mathcal{C}_k \left(f_{1,k}, \ldots, f_{m,k} \right). \tag{17}$$

We discuss one particular and generic example of such $\mathcal{C}_k$ below in Section 4.2. Now we compute:

$$\begin{cases} a_0 = L_0^{-1} f_0, \\ a_k = P_k a_{k-1} + E_k f_k, \quad k = 1, \ldots, n. \end{cases} \tag{18}$$

In the case of just one input image ($m = 1$) the construction (18) reduces to (16).

4.1 Boundary Conditions

At the boundaries of Ω we assume homogeneous Neumann boundary conditions which we discretize in a conservative fashion at Ω_n, e.g. by using bilinear finite elements. The following statements can all be derived from [24]. The boundary conditions inherited by L_k, $0 \leq k < n$, remain homogeneous Neumann ones. All L_k, $0 \leq k \leq n$ have a singular matrix and therefore the L_k^{-1} do not exist. However, systems of type $L_k u_k = g_k$ can still be solved, provided that g_k is in the range of L_k. A sufficient and necessary condition for the latter is proved to be that the sum of elements of g_k vanishes. The said discretization warrants this condition for $k = n$. Further, it is proved that for $k < n$ the f_k defined by (14) inherit the condition. If the condition is satisfied then the algorithm MGD9V [24] is able to solve such singular linear systems iteratively (by multigrid, as explained in Section 2.3). The solution u_k is unique up to a constant (grid-function).

4.2 The Poisson Case

Motivation in 1D. Approximation of second order derivatives of an image grid-function is a popular component of edge detection methods, e.g. Canny [7]. Figure 4 shows an example of an edge profile in one space dimension together with its second derivative. We observe how this edge gives rise to local sources and sinks in the second derivative. This observation provides the basic idea for our fusion method where, loosely formulated, the recombination will be based on choosing the values (+ or $-$) with highest amplitude at geometrically corresponding pixels from a set of input image functions upon which the second derivative operator has been applied. We perform this at each level k and then apply the construction (18). The resulting image combines the edges as observed at all scales of all input images.

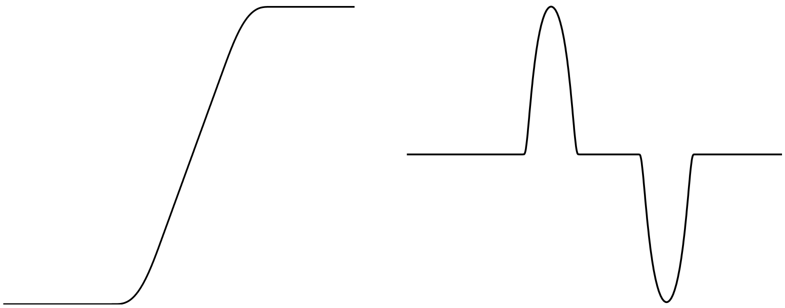

Fig. 4. Edge profile (left) with second derivative (right)

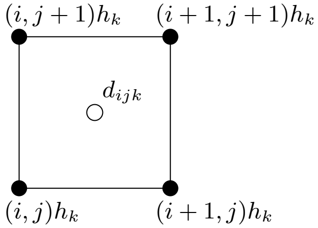

Fig. 5. Cell $C_{ijk} \subset \Omega_k$ with vertices

Generalization. We now have to generalize to two space dimensions. We let L_n be the operator stemming from a discretization by the bilinear finite element method of the Poisson operator $-\Delta$. It can be represented by the 3×3 stencil (or mask)

$$L_n \quad \sim \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & +8 & -1 \\ -1 & -1 & -1 \end{bmatrix} . \tag{19}$$

Both the original operator $-\Delta$ and its above approximation are invariant to rotation. If P_k, $k = 1, \ldots, n$ are prolongations by means of bilinear interpolation then at the coarser grids all L_k produced by (9) turn out to be represented by the stencil (19) as well (but associated with subsequently coarser grids $S(\Omega_k)$, $0 \leq k < n$), see [24].

Fusion and Finite Elements. Considering the definition (12) of E_k we have to ensure that at each level k the f_k resulting from the recombination (17) remains in the range of L_k or else $E_k f_k$ cannot be applied. We achieve this by composing f_k in a finite element manner. The horizontal diffusion operator and vertical diffusion operator are treated separately. Only the contribution of the horizontal operator is described, the contribution of the vertical operator is the analogue. Consider the cell $C_{ijk} \subset \Omega_k$ defined by four indices as indicated in Figure 5.

This cell yields contributions to the stencils of L_k at the four corners, e.g. at gridpoints (i, j) and $(i + 1, j)$ it contributes the respective stencils

$$d_{ijk}\frac{1}{6}\begin{bmatrix} 0 & 1 & -1 \\ 0 & 2 & -2 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad d_{ijk}\frac{1}{6}\begin{bmatrix} -1 & 1 & 0 \\ -2 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

where $d_{ijk} \in \mathbb{R}$ is the diffusion coefficient located at the center of the cell (for now $d_{ijk} = 1$). Such stencils, together with their horizontally mirrored counterparts, add up to stencil (19). When the above stencils are applied on an image grid-function we observe that the contributions at the pixels (i, j) and $(i + 1, j)$ have the same amplitude but opposite sign, hence their sum vanishes.

When fusing a set of m images, for each image grid-function we compute per cell C_{ijk} the contribution, then choose the one from the set of m that is largest

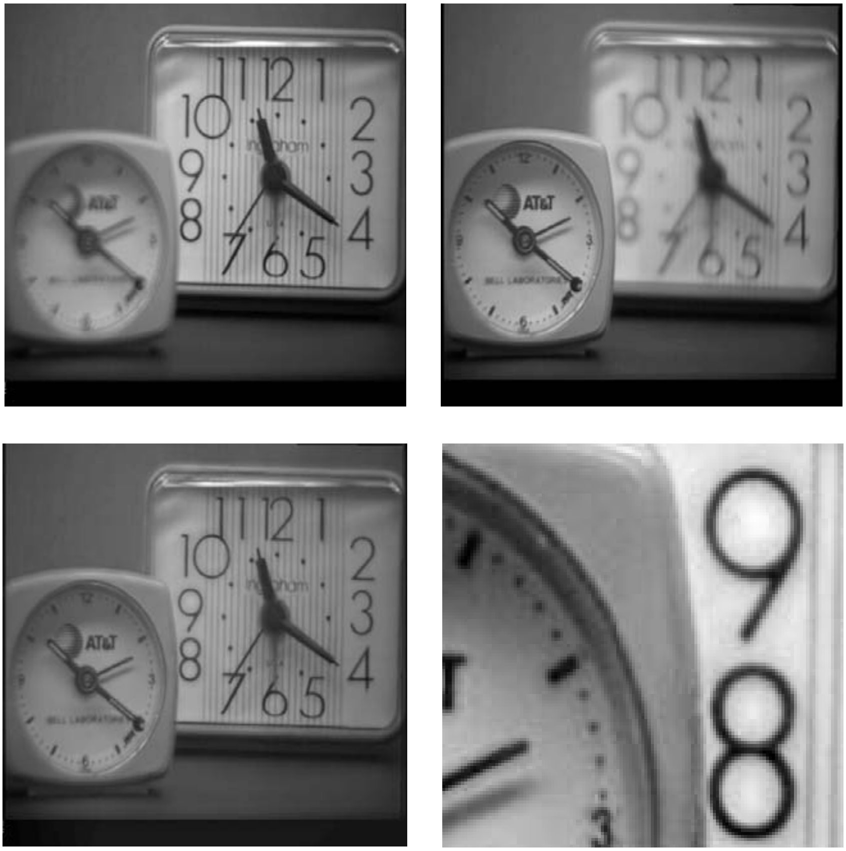

Fig. 6. Top: out-of-focus input images with focus on the right-hand side (left), and with focus on the left-hand side (right). Bottom: fusion of out-of-focus images (left), detail (right)

in absolute value and add this value to the value at pixel (i, j) and the same value but with opposite sign to the value at pixel $(i + 1, j)$. After scanning all cells, the resulting recombined f_k has the desired property.

4.3 Example Fusion Problem

We apply the fusion algorithm of Section 4.2 to two out-of-focus input images, see the top row of Figure 6, the result is to be seen at the bottom row. The quality matches the one obtained by use of the Laplacian pyramid [5] as multiresolution scheme (result not shown).

5 Concluding Remarks

A new multiresolution scheme has been proposed, based on an image transform by a discretized elliptic partial differential operator and use of a multigrid operator, leading to a pyramidal representation. It is shown how this scheme can be applied for image fusion. A single experiment has been added to demonstrate its usefulness. More experiments and an comparison with established methods are in preparation.

The Poisson case as described is just a special case. The framework of the multigrid image transform and multigrid fusion algorithm remains valid if we use the Laplace operator with varying diffusion coefficients instead. An application thereof can be envisaged if we involve segmentation. This is a topic for future research.

References

1. Acton, S.T.: Multigrid Anisotropic Diffusion, IEEE Transactions on Image Processing, Vol. 7, No. 3 (1998)
2. Aubert, G., Kornprobst, P.: Mathematical Problems in Image Processing, Partial Differential Equations and the Calculus of Variations. Applied Mathematical Sciences 147. Springer Verlag, New York (2002)
3. Botta, E.F.F., Dekker, K., Notay, Y., van der Ploeg, A., Vuik, C., Wubs, F.W., de Zeeuw, P.M.: How fast the Laplace equation was solved in 1995. J. Applied Numerical Mathematics 24 (1997) 439–455
4. Brandt, A.: Multi-level adaptive techniques (MLAT) for partial differential equations: ideas and software. In: Rice, J. (ed.): Mathematical Software. Academic Press, New York (1977) 277–318
5. Burt, P.J., Adelson, E.H.: The Laplacian pyramid as a compact image code. IEEE Transactions on Communications, **31** 4 (1983) 532–540
6. Burt, P.J., Kolczynski, R.J.: Enhanced Image Capture through Fusion. Proceedings Fourth International Conference on Computer Vision, Berlin, IEEE (1993)
7. Canny, J.: A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 8 No. 6 (1986) 679–698
8. Hackbusch, W.: Multi-Grid Methods and Applications. Springer, Berlin (1985)

9. Kimmel, R., Yavneh, I.: An algebraic multigrid approach for image analysis, SIAM J. Sci. Comput. Vol. 24, No. 4 (2003) 1218–1231
10. Li, H., Manjunath, B.S., Mitra, S.K.: Multisensor image fusion using the wavelet transform. Graphical Models and Image Processing **57** 3 (1995) 235–245
11. Mallat, S.: A theory for multiresolution signal decomposition: the wavelet representation. IEEE Pattern Analysis and Machine Intelligence Vol. 11, No. 7 (1989) 674–693
12. Osher, S., Sethian, J.A.: Fronts propagating with curvature dependent speed: algorithms based on the Hamilton-Jacobi formulation. J. of Computational Physics **79** (1988) 12–49
13. Piella, G.: A general framework for multiresolution image fusion: from pixels to regions. Information Fusion **9** (2003) 259–280
14. Piella, G.: Adaptive Wavelets and their Applications to Image Fusion and Compression. Ph.D. Thesis, CWI & University of Amsterdam (2003)
15. Sethian, J.A.: Level Set Methods: Evolving interfaces in geometry, fluid mechanics, computer vision, and materials science. Cambridge University Press (1996)
16. Shapira, Y.: Matrix-Based Multigrid: Theory and Applications. Kluwer Academic Publishers, Boston (2003)
17. Simoncelli, E.P., Freeman, W.T.: The steerable pyramid: a flexible architecture for multi-scale derivative computation. Proceedings of the IEEE International Conference on Image Processing, IEEE Signal Processing Society (1995) 444—447
18. Sweldens, W.: The lifting scheme: A construction of second generation wavelets. SIAM J. Math. Anal. **29** 2 (1997) 511–546
19. Terzopoulos, D.: Image Analysis using Multigrid Relaxation Methods, IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 8, No. 2 (1986) 129–139
20. Toet, A.: Hierarchical image fusion. Machine Vision and Applications **3** 1 (1990) 1–11
21. Trottenberg, U., Oosterlee, C.W., Schüller, A.: Multigrid. Academic Press, London (2001)
22. Weickert, J.: Anisotropic Diffusion in Image Processing. Teubner-Verlag, Stuttgart (1998)
23. Wesseling, P.: An Introduction to Multigrid Methods. John Wiley & Sons Ltd., Chichester (1991)
24. de Zeeuw, P.M.: Matrix-dependent prolongations and restrictions in a blackbox multigrid solver. J. Comput. Appl. Math. **33** (1990) 1–27