Multigrid Narrow Band Surface Reconstruction via Level Set Functions

Jian Ye[1], Igor Yanovsky[2,3], Bin Dong[4], Rima Gandlin[5], Achi Brandt[1,6], Stanley Osher[1]

[1] Department of Mathematics, University of California, Los Angeles, CA, USA
[2] Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA
[3] Joint Institute for Regional Earth System Science and Engineering, University of California, Los Angeles, CA, USA
[4] Department of Mathematics, The University of Arizona, Tucson, AZ, USA
[5] Department of Mathematics, Carnegie Mellon University, Pittsburgh, PA, USA
[6] Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel

Abstract. In this paper we propose a novel fast method for implicit surface reconstruction from unorganized point clouds. Our algorithm employs a multigrid solver on a narrow band of the level set function that represents the reconstructed surface, which greatly improves computational efficiency of surface reconstruction. The new model can accurately reconstruct surfaces from noisy unorganized point clouds that also have missing information.
Keywords. Level set, multigrid method, point cloud, surface reconstruction.

1 Introduction

The field of surface reconstruction from scattered point clouds has been developing rapidly in the past decade. It is a challenging problem since point clouds lack ordering information and connectivity, and are usually noisy. There are two ways of representing the reconstructed surfaces: explicit and implicit. Explicit representation usually gives the exact location of a surface in a physical domain, while implicit representation defines the surface as the (zero) level set of some scalar function. Common explicit representations include parametric surfaces [1, 2] and triangulated surfaces [3–7]. Implicit surfaces are most frequently represented using level set functions, typically signed distance functions [8], as well as some more recent ones [9, 10].

One of the traditional approaches for implicit surface reconstruction is the use of radial basis functions representation [11]: $s(\boldsymbol{x}) = p(\boldsymbol{x}) + \sum_{i=1}^{n} \lambda_i \phi(|\boldsymbol{x} - \boldsymbol{x_i}|)$, where p is a polynomial, ϕ is a global smooth function that allows fast summation, e.g. $\phi(\boldsymbol{r}) = \boldsymbol{r}$, and $(\boldsymbol{x_1}, ..., \boldsymbol{x_n})$ is a set that includes the N given surface points and a comparable number of off-surface points at each of which $s(\boldsymbol{x_i})$ is prescribed as an estimated distance from $\boldsymbol{x_i}$ to the surface. The iterative solver

for computation of the coefficients λ_i and the polynomial p requires $C_1 N \log N$ computer operations, where C_1 is a very large constant.

Another well-known method was introduced in [12, 13], where the authors constructed a weighted minimal surface-like model. The reconstructed surface is represented as the zero level set of a scalar function and the proposed energy functional is minimized by solving a nonlinear partial differential equation (PDE) (see [12, 13] for details). However, solving such PDE requires small time steps and hence longer computational time. Furthermore, the energy functional proposed in [12, 13] is nonconvex, which makes the result sensitive to the initialization and noise. Therefore, in order to avoid local minimizers and reduce computation time, one should start from an initial surface which is very close to the given point cloud. More recently, the authors in [14, 15] extended the work in [12, 13], proposing new models and solving the surface reconstruction problem using state-of-the-art optimization algorithms

All the aforementioned level set based surface reconstruction methods aimed at solving some nonsmooth (and often nonconvex) optimization problem. One of the main benefits of solving such nonsmooth problems is to preserve very sharp features, such as edges, of the points clouds in the reconstructed surfaces. However, the nonsmoothness also inevitably increases the difficulty of solving the problems efficiently. In practice, many geometric objects to be reconstructed from their point samples do not have very sharp features. Therefore, for these cases, solving nonsmooth optimization problems is not necessary. In this paper, we propose a differentiable variational model for surface reconstruction problems, which is solved very efficiently by using multigrid techniques on a narrow band of the level set function that represents the reconstructed surface. In addition, important features of the point clouds are also well preserved in the reconstructed surfaces using our proposed algorithm.

2 Proposed Model

Given a data set $\{\boldsymbol{x}_l\}_{l=1,\ldots,N} \subset \mathbb{R}^{\dim}$, i.e. a set of points with dim = 2 or 3, we seek a function ϕ that is close to zero on this set and smooth elsewhere (see [16]). For this purpose, we consider the following energy functional:

$$E(\phi) = \int G(\phi(x))dx + \sum_{l=1}^{J} \beta_l (P_l \phi)^2, \tag{1}$$

where the projection operator P_l is some local averaging operator defined as

$$P_l \phi = \int p_l(x)\phi(x)dx, \quad \int p_l(x)dx = 1. \tag{2}$$

The first term in (1) is the regularization term which imposes smoothness on the level set function ϕ. The second term is the fidelity term which forces the zero level set of ϕ align with the data set. If the data is uniformly distributed

and the surface is well-resolved in some region, we set

$$G(\phi(x)) = |\nabla\phi(x)|^2. \tag{3}$$

Otherwise, in regions that lack points, more sophisticated regularization is required, for instance, defining $G(\phi(x)) = |\nabla\cdot\nabla\phi(x)|^2$. Furthermore, special regularization is needed for the anisotropic case when the coefficient function varies in different directions. For example in $\mathbb{R}^2$, when the coefficient function $a_1(x,y)$ in the x-direction differs from the coefficient function $a_2(x,y)$ in the y-direction, we set

$$G(\phi(x,y)) = a_{11}|\phi_x|^2 + a_{12}|\phi_x\phi_y| + a_{22}|\phi_y|^2 = \alpha_1|\phi_\xi|^2 + \alpha_2|\phi_\eta|^2$$

with some proper change of variables $(x,y) \to (\xi,\eta)$. In this paper, we will focus on the isotropic regularization (3). However, the proposed method can be easily extended to other cases as well.

The values of the weight function β_l should, in general, depend on the accuracy of data points, the density of data set, and the curvature of surface to be reconstructed. For simplicity, a fixed β is used for all data points in this paper.

We now provide the detailed formulation of the energy $E(\phi)$ and the projection operator P_l in the discrete setting. We first consider the two dimensional case. The discretization of the projection operator (2) takes the following general form:

$$P_l\phi = \sum_{k,n} p^l_{k,n}\phi_{k,n}, \quad \sum_{k,n} p^l_{k,n} = 1. \tag{4}$$

The energy functional (1) is discretized as

$$E(\phi) = \sum_i \sum_j \Big(\frac{\phi_{i+1,j}-\phi_{i,j}}{h}\Big)^2 + \Big(\frac{\phi_{i,j+1}-\phi_{i,j}}{h}\Big)^2 + \sum_l \beta_l[P_l\phi]^2. \tag{5}$$

Then at the grid point (i,j), the Euler-Lagrange equation is given by

$$\frac{1}{2}\frac{\delta E}{\delta\phi_{i,j}} = -\frac{\phi_{i+1,j}+\phi_{i-1,j}+\phi_{i,j+1}+\phi_{i,j-1}-4\phi_{i,j}}{h^2} + \sum_{l=1}^{J}\beta_l p^l_{i,j}P_l\phi = 0, \tag{6}$$

where J is the total number of neighboring points of grid point (i,j). Operator P_l is a linear interpolation operator, and $P_l\phi$ represents the interpolated value of function ϕ at a point $\boldsymbol{x}_l$. Fig. 1 shows the l-th point inside a grid cell associated to (i,j), with location determined by r_1 and r_2 $(0 \le r_1, r_2 < h)$. The interpolation coefficients are:

$$\begin{aligned} &p^l_{i,j} = \frac{(h-r_1)(h-r_2)}{h^2}, \quad p^l_{i,j+1} = \frac{(h-r_1)r_2}{h^2}, \\ &p^l_{i+1,j} = \frac{r_1(h-r_2)}{h^2}, \quad p^l_{i+1,j+1} = \frac{r_1r_2}{h^2}, \\ &P_l\phi = p^l_{i,j}\phi_{i,j} + p^l_{i,j+1}\phi_{i,j+1} + p^l_{i+1,j}\phi_{i+1,j} + p^l_{i+1,j+1}\phi_{i+1,j+1}. \end{aligned} \tag{7}$$

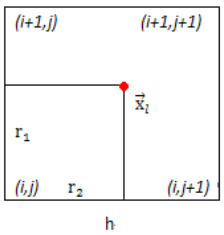


Fig. 1. The illustration of 2D interpolation operator. Here r_1 is the displacement in vertical direction and r_2 is the displacement in horizontal direction.

In three dimensions, the corresponding Euler-Lagrange equation at a point (i, j, k) is given as

$$\frac{1}{2}\frac{\delta E}{\delta \phi_{i,j,k}} = -\frac{\phi_{i+1,j,k} + \phi_{i-1,j,k} + \phi_{i,j+1,k} + \phi_{i,j-1,k} - 6\phi_{i,j,k}}{h^3} + \sum_{l=1}^{J} \beta_l p_{i,j}^l P_l \phi = 0. \tag{8}$$

The natural choice for interpolation operator in three dimensions is trilinear interpolation. In the following formulation, r_1 and r_2 are defined as above, with r_3 denoting the displacement in the z dimension. The interpolation coefficients in three dimensions are given as:

$$\begin{aligned} p_{i,j,k}^l &= \frac{(h-r_1)(h-r_2)(h-r_3)}{h^3}, & p_{i,j+1,k}^l &= \frac{(h-r_1)r_2(h-r_3)}{h^3}, \\ p_{i+1,j,k}^l &= \frac{r_1(h-r_2)(h-r_3)}{h^3}, & p_{i+1,j+1,k}^l &= \frac{r_1 r_2(h-r_3)}{h^3}, \\ p_{i,j,k+1}^l &= \frac{(h-r_1)(h-r_2)r_3}{h^3}, & p_{i+1,j+1,k+1}^l &= \frac{r_1 r_2 r_3}{h^3}, \\ p_{i+1,j,k+1}^l &= \frac{r_1(h-r_2)r_3}{h^3}, & p_{i,j+1,k+1}^l &= \frac{(h-r_1)r_2 r_3}{h^3}, \end{aligned} \tag{9}$$

In the next section, we will describe an efficient numerical implementation for solving (6) and (8).

3 Numerical Implementation

In this section, we consider two dimensional case and describe the process of obtaining a numerical solution for equation (6). The treatment for three dimensional case (8) is similar.

3.1 Initialization

In order to solve equation (6) numerically, we define a suitable computational domain. First, given the coordinates of the data $\{\boldsymbol{x}_l\}$, we find the smallest rect-

angular box that contains the data. Since extra space should be allocated at the boundaries of the computational domain, we extend the rectangular box by some factor ρ. The typical choice for extension is $\rho = 1.2$. In case data set has a large hole (i.e. a large region with missing information), the value ρ is increased.

To make the algorithm efficient, we can restrict our computations within a narrow band containing the data. To construct the narrow band, we first compute the unsigned distance function $d(x)$ to the data set S by solving the following Eikonal equation (See equation (10))

$$|\nabla d(x)| = 1, \text{ on } \Omega \setminus \Gamma \quad \text{and} \quad d(\Gamma) = 0. \tag{10}$$

For the boundary conditions of (10), if the data point is not on the grid, we set the unsigned distance function d to zero at the nearest neighboring grid point to the data point in consideration. We then solve equation (10) using fast sweeping method [17, 18], an efficient algorithm with a computational cost of $O(N)$. The narrow band is obtained by thresholding unsigned distance function at value $\epsilon = \frac{1}{2} m_h h$, where h is the mesh size and m_h is the band width. Here, we denote $\Omega_1 = \{x : d(x) < \epsilon\}$ to be the set of grid points within the narrow band.

Once the narrow band is obtained, we need to find the grid points for the outer and inner boundaries in order to set up the boundary conditions for (6). Outer and inner boundary grid points are denoted as Γ_{in} and Γ_{out}, respectively, and are obtained using the Breadth-First Search (BFS) algorithm. First, we take a thin narrow band Ω_2, which is directly connected to Ω_1, such that the band width is equal to one grid point. Specifically, $\Omega_2 = \{x : \epsilon \leq d(x) \leq \epsilon + h\}$. Second, we select an arbitrary outer boundary grid point $(i, j) \in \Omega_2$. This can be easily done by taking the first grid point in Ω_2 since the outer boundary of the narrow band is usually reached while traversing the grid points. This grid point is then assigned to Γ_{out}. Third, we use BFS algorithm to find all the connected grid points among $(i \pm 1, j), (i, j \pm 1)$ which are assigned to Γ_{out}. The algorithm stops when there are no new grid points to be added into Γ_{out}. The complementary set of Γ_{out} is the inner boundary $\Gamma_{\text{in}} = \Omega_2 / \Gamma_{\text{out}}$.

If the data has spurious noise, e.g. contains points far from the meaningful data points, the noise and the data set would be disconnected. In this case, all connected components are obtained using the BFS algorithm as described above. We first select one starting point and find the maximally connected component. Then, we select another point in the complementary set and find another maximally connected component. The process is repeated until all data points are traversed once. Since spurious noise is usually composed of relatively few data points, we let Γ_{out} and Γ_{in} be the largest and second largest connected components, respectively.

Once exterior and interior boundary grid point sets Γ_{in} and Γ_{out} are obtained, we can specify boundary conditions for $\phi(x)$. For the exterior boundary Γ_{out}, we set $\phi(x) = d(x)$; for the interior boundary Γ_{in}, we set $\phi(x) = -d(x)$; and for the points within the narrow band, we simply set $\phi(x) = 0$.

Fig. 2 shows the initialization of the narrow band associated with the given data set. The red dots and black dots enclose the blue data points with the equal

distance $\epsilon = 5h$. The unsigned distance function at the red outer boundary and the black inner boundary points is equal to $5h$ and $-5h$, respectively. The grid points inside the narrow band are all set to zero.

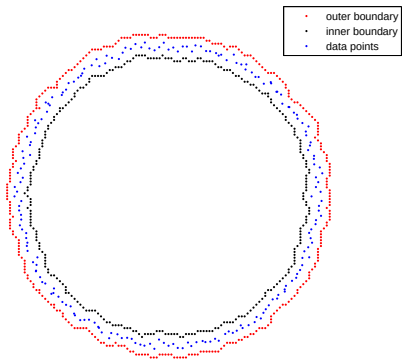


Fig. 2. Illustration of inner and outer boundaries.

3.2 Multigrid Narrow Band Solver

The problem defined by (6) with boundary conditions is a well-posed elliptic problem and can be efficiently solved using multigrid method. The Full Approximation Scheme (FAS) [19] solves equation (6) on multiple grids, starting with the coarsest and finishing at the finest grid. On coarser levels, a single cell may contain multiple data points. Since high resolution results are not required on coarser grids, mean coordinates of all data points in a given cell can be used to represent all such points for the purpose of calculating the interpolation operator P_l^l. After the equation is solved on a coarser level using a few Gauss-Seidel relaxations [19], the obtained solution is linearly interpolated onto the finer level. At the finest level, all data points within each cell are considered to be carrying equal weights. The solution obtained on the finest level is the desired final reconstruction. We usually use three-level construction method with the band width $m_h h$. Here, $m_h = 5$, and h is the mesh size for the corresponding level.

We are now ready to give an algorithm for the Multigrid Narrowband Surface Reconstruction:

4 Experimental results

In this section, we show the two and three dimensional results obtained using the proposed Multigrid Narrow Band surface reconstruction model to solve (6) and (8). In our numerical examples, we found that $\beta = 0.5$ produces desirable

Algorithm 1 Multigrid Narrow Band Surface Reconstruction

1: Given a set $\{\boldsymbol{x}_l\}_{l=1,\ldots,N}$, solve Eikonal equation (10) to find $d(x)$.
2: Find a narrow band Ω_1 with band width ϵ to enclose the data set.
3: Find the outer boundary Γ_{out} and inner boundary Γ_{in} of the narrow band.
4: Set $\phi(x) = \epsilon$ on the outer boundary and $\phi(x) = -\epsilon$ on the inner boundary.
5: Denote l to be the coarsest level.
6: Solve equation (6) on level l within narrow band. If l is the finest level, then stop.
7: Linearly interpolate the solution from level l to a finer level $l + 1$.
8: Set $l := l+1$ and go to Step 6; or terminate the program if the finest grid is reached.

results for most data sets. The value of β may change somewhat with changes in noise level and density for a given data set. Fig. 3 shows a two-dimensional point cloud and multigrid surface reconstruction results at four consecutive steps on different levels. The reconstruction captures additional details as the grid is refined, with the finest level having the most information.

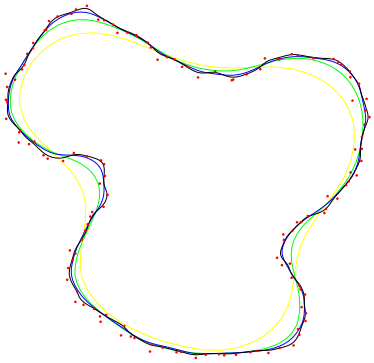


Fig. 3. Contours reconstructed on different levels, from coarsest to finest, are displayed in yellow, green, blue, and black colors.

Fig. 4 shows a two-dimensional numerical result for a man-made hand with Gaussian noise. The hand shape has long and thin concave regions which are difficult to reconstruct using traditional methods. However, our method recovers the detailed information. We observe some inaccuracy at the ends of the fingers, which may be attributed to the use of uniform β for high curvature places.

Next we consider some three dimensional examples. Fig. 5 shows 3D multigrid surface reconstruction results for "bunny" point cloud on different levels. The difficulty of reconstructing this point cloud lies in the presence of several holes. We observe that finer features are recovered as the mesh size becomes smaller, and the holes are filled automatically. On $257 \times 257 \times 201$ mesh, the total computational time to obtain the final result on 2.93 GHz Intel Core 2 Duo CPU is 6.5 seconds (see Table 1).

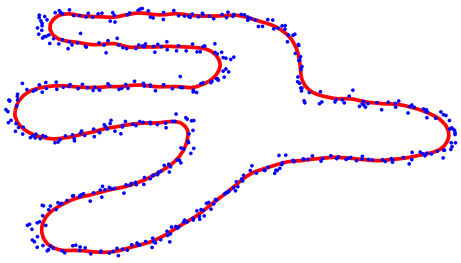


Fig. 4. Contour reconstruction of a man-made hand with Gaussian noise, where the noisy point cloud (in blue) and the reconstructed shape (in red) are displayed.

Table 1. Computational times for 3D data sets.

data set	# points	grid size	time (in seconds)
bunny	35,947	257x257x201	6.5
dragon	437,645	301x213x133	7.2
buddha	144,647	149x365x149	6.7

Fig. 6(a) shows 3D surface reconstruction result from "dragon" point cloud. Since this data set contains numerous regions with missing information (i.e. small holes in the point cloud), it is difficult to reconstruct the corresponding surface. The proposed model is successful in capturing fine details and filling the holes in the surface.

Fig. 6(b) shows 3D surface reconstruction result from Buddha point cloud. Even though this data set contains no holes, it has small bridges. We observe that all fine features are captured well using the proposed surface reconstruction model.

5 Conclusion

The two and three dimensional curve/surface reconstruction results presented in this paper demonstrate that our method is among the fastest surface reconstruction methods. Furthermore, the proposed method is robust to noise and can easily recover surfaces from point clouds with missing information. Important details and features of the point clouds are also well preserved in the reconstructed surfaces.

Acknowledgements

This work was supported in part by NIH GRANT, P20 MH65166, NSF GRANT, DMS-0714807, and NIH GRANT, U54 RR021813. The research of Igor Yanovsky was carried out in part at the University of California, Los Angeles, and in part

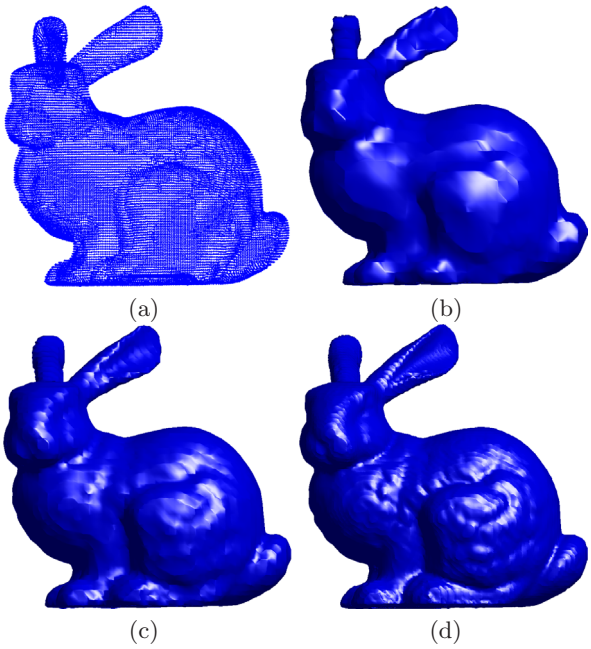
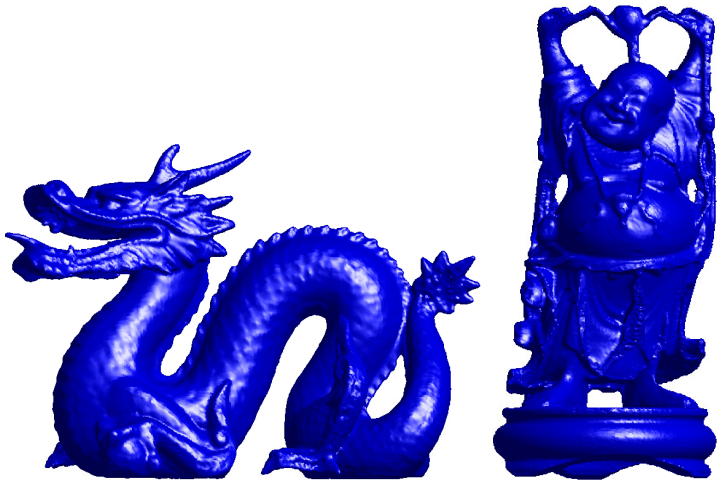


Fig. 5. Three dimensional reconstruction of "bunny". (a) Original point cloud. The results are obtained on (b) 65x65x51, (c) 129x129x101, and (d) 257x257x201 grids.

at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

1. Rogers, D.: An Introduction to NURBS. Morgan Kaufmann (2003)
2. Piegl, L., Tiller, W.: The NURBS book. Springer-Verlag, Berlin, Germany (1996)
3. Amenta, N., Bern, M., Eppstein, D.: The crust and the β-skeleton: Combinatorial curve reconstruction. Graphical Models and Image Processing **60** (1998) 125–135
4. Amenta, N., Bern, M., Kamvysselis, M.: A new Voronoi-based surface reconstruction algorithm. In: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, ACM New York, NY, USA (1998) 415–421
5. Boissonnat, J.: Geometric structures for three dimensional shape reconstruction. ACM Trans. Graphics 3 (1984) 266–286
6. Edelsbrunner, H.: Shape reconstruction with Delaunay complex. Lecture Notes in Computer Science (1998) 119–132
7. Edelsbrunner, H., Mucke, E.P.: Three dimensional α shapes. ACM Trans. Graphics 13 (1994) 43–72
8. Osher, S., Fedkiw, R.: Level set methods and dynamic implicit surfaces. Springer (2003)
9. Leung, S., Zhao, H.: A grid based particle method for moving interface problems. Journal of Computational Physics **228** (2009) 2993–3024



(a) (b)

Fig. 6. Three dimensional reconstruction of "dragon" (a) and "buddha" (b).

10. Ruuth, S., Merriman, B.: A simple embedding method for solving partial differential equations on surfaces. Journal of Computational Physics (2007)
11. Carr, J., Fright, W., Beatson, R.: Surface interpolation with radial basis functions for medical imaging. IEEE Transactions on Medical Imaging **16** (1997) 96–107
12. Zhao, H., Osher, S., Fedkiw, R.: Fast surface reconstruction using the level set method. In: Variational and Level Set Methods in Computer Vision, 2001. Proceedings. IEEE Workshop on, IEEE (2002) 194–201
13. Zhao, H., Osher, S., Merriman, B., Kang, M.: Implicit and non-parametric shape reconstruction from unorganized points using variational level set method. Computer Vision and Image Understanding **80** (2000) 295–319
14. Goldstein, T., Bresson, X., Osher, S.: Geometric applications of the split bregman method: Segmentation and surface reconstruction. Journal of Scientific Computing **45** (2010) 272–293
15. Ye, J., Bresson, X., Goldstein, T., Osher, S.: A Fast Variational Method for Surface Reconstruction from Sets of Scattered Points. CAM Report **10-01** (2010)
16. Gandlin, R.: Multigrid solvers for inverse problems. Ph.D. thesis, Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot, Israel (2004)
17. Zhao, H.: A fast sweeping method for eikonal equations. Mathematics of computation **74** (2005) 603–628
18. Tsai, Y., Cheng, L., Osher, S., Zhao, H.: Fast sweeping algorithms for a class of Hamilton-Jacobi equations. SIAM journal on numerical analysis **41** (2004) 673–694
19. Brandt, A.: Multigrid techniques. Ges. für Mathematik u. Datenverarbeitung (1984)