

An Algebraic Multigrid Approach for Image Analysis

RON KIMMEL IRAD YAVNEH*
Department of Computer Science
Technion–Israel Institute of Technology
Technion City, Haifa 32000, Israel

May, 2002

Abstract

We apply a new algebraic multigrid method for solving computer vision problems with constraints. As particular examples we solve the “shape from photometric stereo” and “image binarization” problems. A variational formulation is applied to the problem of shape reconstruction from three or more images of an object with the same viewing direction and different lighting conditions, supplemented by some pointwise height constraints. In order to obtain a smooth reconstruction, we use a weight-function that is singular at the constrained points, resulting in an elliptic equation with singular coefficients, which is solved efficiently by the algebraic multigrid algorithm. As a second example a similar technique is applied to construct a threshold surface which interpolates between values at centers of edges. This surface is then used for image binarization.

1 Introduction

Many techniques of image analysis require solving elliptic boundary-value problems, often emerging from the minimization of some functional. For constant-coefficient problems on rectangular domains Fourier methods are very efficient, but for varying coefficients or more general domains multigrid iterative methods are probably the most effective available solution technique (see description and references in Section 2 below.) Multigrid methods were proposed in the 1960’s and established as efficient solvers for elliptic boundary-value problems in the 1970’s. They were first applied to image analysis by Terzopoulos (e.g., [17]). More recent efficient applications using classical multigrid structures include [21], which applies a fairly standard multigrid algorithm for several image processing applications; [14], where powerful fast-transform preconditioned conjugate gradient smoothers are employed to handle difficult (and dense) differential-convolution equations by multigrid algorithms; [11], where a linear coupled system of partial differential equations is solved repeatedly within an iterative algorithm for digital image matching.

*The second author was supported in part by the Center for Applied Scientific Computing of the Lawrence Livermore National Laboratory and the Israeli Ministry of Science and the Arts.

It is well-known that classical (also called “geometric”) multigrid methods are not robust with respect to discontinuous coefficients and singular boundaries (particularly small “holes” in the domain.) Such problems usually require specialized robust multigrid methods. Such situations appear in certain problems of image analysis and computer vision, particularly if the mathematical model which yields the boundary-value problem is supplemented by pointwise constraints where the (exact or approximate) solution and/or its derivatives are prescribed. In this paper we introduce an efficient and robust multigrid method and apply it to two image analysis techniques of this sort where such robustness is essential for obtaining fast asymptotic convergence rates.

A major part of computer vision deals with the problem of shape reconstruction from two dimensional projections of the real world onto a camera. These projections, which we usually recognize as two dimensional images, serve as the key for the shape reconstruction, where the whole family of shape reconstruction problems is known as ‘shape from stereo/sequence/shading/structured-light/etc.’ One distinguished member of this shape reconstruction family is the ‘shape from photometric stereo’ problem which we use as our first example in this paper. In this problem the camera location is fixed, and several images are obtained with different lighting conditions. A simple model for the way the shape reflects the light is the Lambertian reflectance model, according to which the gray level observed is proportional to the inner product of the light source direction and the surface normal. The proportionality ratio depends on the object properties, known as *albedo*, and the light source intensity, which can be normalized.

Given three images I_1, I_2, I_3 , of the same object taken with three different lighting directions l_1, l_2, l_3 , the following relation holds for a Lambertian reflectance model,

$$I_i = \rho \langle l_i, N \rangle,$$

where, $i \in \{1, 2, 3\}$, ρ is the albedo, and N is the normal to the surface, $z(x, y)$, given by

$$N = \frac{(-z_x, -z_y, 1)}{\sqrt{1 + z_x^2 + z_y^2}}.$$

Using the Lambertian reflectance model, the approximate surface gradient, $(p, q)^T \approx \nabla z$, can be extracted easily from the images. The question we deal with is how to integrate back the surface $z(x, y)$ from its approximate gradient vector field $(p, q)^T$, keeping in mind that there are errors in the model and in the measurements. A natural approach is to adopt a variational formulation, plugging the given gradient values into a global measure for the reconstructed surface z and searching for the surface which minimizes the functional

$$\int_{\Omega} w(x, y) \|(p, q)^T - \nabla z\|^2 dx dy, \quad (1)$$

where $w(x, y)$ is some positive weight function. The resulting Euler-Lagrange equation is

$$\begin{aligned} \nabla \cdot [w(x, y)(p - z_x, q - z_y)^T] &= 0, & (x, y) &\in \Omega, \\ \nabla z \cdot \mathbf{n} &= (p, q) \cdot \mathbf{n}, & (x, y) &\in \partial\Omega, \end{aligned} \quad (2)$$

where \mathbf{n} is the outwards normal to the boundary of the image, $\partial\Omega$.

For the simple choice of $w \equiv 1$, we obtain the Poisson problem, which is easy to solve by standard numerical methods. But, since the model and measurements may contain significant errors, we are particularly interested in the case where we can assume some additional knowledge about the surface profile, such as the surface height at specific coordinates or along a curve. This introduces additional constraints on the value of $z(x, y)$ at certain points or regions in Ω . This problem was first studied by Horowitz and Kiryati [13] (henceforth HK). They assumed that the height values were given at some coordinates, and found that these constraints resulted in an appreciable improvement in the solution of the reconstruction problem, especially when the data were noisy. HK suggest two approaches for dealing with the problem. One approach is to minimize (1) subject to constraints on z at selected points. This requires using a weight function with singularities at the constrained points. Suppose we have a constraint at the point (x_i, y_j) . Let $r_{i,j} = \|(x, y) - (x_i, y_j)\|_2$ denote the Euclidean distance from the constrained point. Then, in order to maintain k continuous derivatives of the reconstructed shape at (x_i, y_j) , we must have $w(x, y) \sim r_{i,j}^{-\alpha}$ in a neighborhood of (x_i, y_j) , with $\alpha > k$.

A second approach offered by HK is to split the problem into two parts, solving the unconstrained problem with $w \equiv 1$, and then adding a smooth function obtained by interpolation between the constrained points, such that the sum will satisfy the constraints. The interpolation in the second step is performed using radial basis functions. Thus, the computational complexity of this step is $O(cn)$, where c is the number of constrained points and n is the number of variables. This is unsatisfactory when c is large, particularly when the height is constrained on a curve or a surface, although it is quite efficient for small c . When c might be large, we prefer to adopt the first approach. Accordingly, we impose pointwise constraints on z and discretize (2) at all unconstrained points. Solving the resulting linear system of equations requires special methods to obtain high computational efficiency. The difficulty stems from the singularities in w and the holes in the domain at the constrained points, where z is prescribed and (2) is not imposed. (This, in turn, results from the minimization of (1) subject to the constraints.) Our goal is to solve the problem to the level of discretization errors in $O(n)$ operations independently of c .

2 The Algebraic Multigrid Approach

The computational problem of solving a linear system of equations arising from the discretization of an elliptic partial differential equation is well-studied. Here, however, there are complications due to the constraints and the singular form of the weight function, w . For a very small number of constraints, suitably modified “standard” methods based on the Fast Fourier Transform (FFT) algorithm or “classical” multigrid methods may be efficient. But the additional work required for c constrained points will normally be at least $O(cn)$. This renders such methods inefficient if there are more than just a few constrained points, and particularly if there are constrained curves or surfaces. Furthermore, the so-called pyramidal method commonly used has a very slow n -dependent asymptotic convergence rate even for the simple Poisson problem. We therefore adopt an algebraic multigrid (AMG) approach, which achieves fast convergence independently of the number of constraints. There is a computational overhead involved, but it too is independent of the number of constraints and depends linearly on n .

The concept of exploiting several different grids for accelerating iterative solutions of dis-

cretized partial differential equations (PDEs) was explored in the 1960's, and the first practical algorithms were formulated and implemented in the 1970's (see [2] and the historical notes therein.) A fine elementary introduction to multigrid computational methods, including a chapter on AMG, is [5]. A comprehensive source for the practice and practical theory of multigrid methods, including many applications and an excellent introduction to AMG (Stüben, Appendix A) is [20].

The basic idea of multigrid methods is to employ a sequence of progressively coarser grids, which geometrically include the fine-grid domain, to accelerate some basic iterative solver. The latter is usually a classical relaxation method, such as damped Jacobi, Gauss-Seidel or SOR. In this framework, the relaxation is required only to *smooth* the error relative to the computational grid, where the error is defined as the difference between the exact solution to the discrete equations and the approximate solution at any stage of the algorithm, and “error smoothness” can be loosely defined as the property which allows accurate approximation of the error on the next-coarser grid employed. Hence, the relaxation operator is usually referred to as a “smoother”. Once the error is smooth relative to the current grid, it is approximated on the next-coarser grid using an appropriate coarse-grid correction equation, and the resulting approximate correction is interpolated back to the fine grid and used to correct the fine-grid approximation. The coarse-grid problem is solved approximately by a similar process using a still coarser grid, and so on recursively. The efficiency of recursion is due to the fact that an error which is smooth relative to a given grid is less smooth relative to a coarser grid, so the smoother can be applied effectively once again. The process described here is commonly called the (coarse-grid) correction scheme.

Most multigrid algorithms employ the same basic structure, and the differences are mainly in the choice of the particular components, e.g., the smoother and the inter-grid transfer operators. The latter are the *prolongation* (or interpolation), which is employed to transfer grid data to the next-finer grid, and the *restriction*, which is used for fine-to-coarse grid transfer. For elliptic partial differential operators with smoothly varying coefficients, simple methods such as linear interpolation for the prolongation, and local-averaging or injection for the restriction, are most efficient. But discontinuous coefficients and local constraints as in the present problem require special handling. The problem of discontinuous coefficients in multigrid solvers was first studied in [1], and several different approaches were later developed (e.g., [8, 9]). Further advancements which widened the scope and robustness of multigrid methods were brought about by AMG methods, introduced in [4] and developed in [15]. Subsequent developments have been numerous and branched off in many directions (see, e.g., Stüben [20, Appendix A] and references therein). AMG methods allow solution of problems discretized on unstructured grids, including non-PDE applications. Of course, a significant computational overhead is required to obtain this extra flexibility. In the problems studied here we do not require such generality, and this allows us to avoid many of the complications associated with unstructured meshes. Hence, also, we do not try to list or compare different AMG approaches, most of which are aimed at handling more general settings than we consider here, and are therefore understandably more costly.

2.1 The multigrid components

Here we present the details of the AMG algorithm. Let the linear system resulting from the finite-difference discretization be

$$A\bar{z} = \bar{f},$$

where A is a matrix of size $n \times n$, and \bar{z} and \bar{f} are, respectively, an unknown and a given n -vector. We discretize with the standard second-order finite difference discretization, first eliminating constrained variables. The resulting discretization stencil at point (x_k, y_ℓ) is given by

$$\begin{bmatrix} \cdot & -w(x_k, y_\ell + h/2) & \cdot \\ -w(x_k - h/2, y_\ell) & \Sigma_w & -w(x_k + h/2, y_\ell) \\ \cdot & -w(x_k, y_\ell - h/2) & \cdot \end{bmatrix}, \quad (3)$$

where

$$\Sigma_w = w(x_k, y_\ell + h/2) + w(x_k, y_\ell - h/2) + w(x_k + h/2, y_\ell) + w(x_k - h/2, y_\ell),$$

and h is the meshsize. If a neighboring point is constrained, then the corresponding element is eliminated from the stencil (and a corresponding term is subtracted from the right-hand side of the equation.) Thus, if point $(x_k, y_{\ell+1})$ is constrained, then the element $w(x_k, y_\ell + h/2)$ is eliminated from the stencil above, but the stencil is otherwise unaffected. In particular, the diagonal term is unchanged. Since $w > 0$, the equations at points adjacent to constraints are therefore strictly diagonally dominant. At the constrained points themselves, it is convenient to retain variables (with trivial diagonal equations) in order to maintain a regular rectangular grid. The resulting discretization matrix, $A = \{A_{i,j}\}$, is positive definite and diagonally dominant. In fact, for every row $i = 1, \dots, n$, we have

$$A_{i,i} > 0, \quad A_{i,j} \leq 0 \text{ for } j \neq i, \quad A_{i,i} \geq -\sum_{j \neq i} A_{i,j},$$

where the last inequality is strict at constrained points and their nearest neighbors. Furthermore, the discretized operator is isotropic. For such matrices, simple Gauss-Seidel or damped Jacobi relaxation are known to provide excellent smoothing.

For the discussion of the coarse-grid correction operators it suffices to consider just two grids—the “fine” grid, which is the set of variables z_i^f , $i \in \{1, \dots, n^f\}$, and the “coarse” grid, z_j^c , $j \in \{1, \dots, n^c\}$. The extension to a *multi-grid* algorithm is obtained by the usual recursion resulting in a so-called V-cycle. The coarse-grid variables are chosen to coincide with a subset of the fine-grid points. Accordingly, we partition the set of fine-grid indices as follows:

$$\{1, 2, \dots, n^f\} = C \cup F, \quad (4)$$

where C denotes the set of indices of fine-grid variables which coincide with coarse-grid variables, and $F \cap C = \emptyset$.

Given the matrix A^f associated with the fine-grid equations, and the partition (4), we need to define the prolongation matrix, P , the restriction matrix, R , and the coarse-grid equation matrix, A^c . We adopt the standard choices for symmetric matrices: $R = P^T$ for the restriction, and $A^c = RA^fP$ for the (Galerkin) coarse-grid matrix. A more general discussion on how to define the partition (4) and how to treat nonsymmetric matrices will appear in a later paper; these issues are outside the scope of the present application. It remains now to define the prolongation.

2.1.1 Prolongation.

The prolongation approximates the error at F points, given (approximate) values of the error at C points. Matrix-dependent prolongations employed in multigrid methods are generally based on some local approximate solution of the homogeneous fine-grid problem at F points, given fixed values at nearby C points. We do this by means of one or a few local sweeps of F -relaxation. These are simply relaxation sweeps in which C points are skipped, and therefore C -variables remain unchanged (see also Stüben, [20, Appendix A]). If the coarse grid points are chosen properly, F relaxation must converge very fast [3]. A “symmetric” relaxation should be preferred to avoid directional biases which might impair convergence. So Jacobi relaxation should be preferred to lexicographically ordered Gauss-Seidel, for example. In the structured-grid case tested below, we found Black-Red relaxation (see below) to be even more effective.

The operator P is a matrix of size n^f by n^c , where $P_{i,j}$ is the weight corresponding to the contribution of the coarse-grid variable z_j^c to the fine-grid variable z_i^f . For every coarse-grid point $j \in \{1, \dots, n^c\}$, let $i_j \in C$ denote the fine-grid point which coincides with it. We prescribe a local interpolatory set (prolongation stencil), $\Gamma_j \subseteq F \cup \{i_j\}$, which is the set of all points to which the variable z_j^c contributes in the prolongation. In particular, $i_j \in \Gamma_j$, that is, the coarse grid point j contributes to the C point which coincides with it but to no other C point. We set $P_{i,j} = 0$ for $i \notin \Gamma_j$ and construct the remainder of the matrix P by the following two-stage procedure (see Fig. 1).

PROCEDURE 1

Tentative Prolongation Operator

Define a tentative prolongation matrix, $P^{(tentative)}$, of size $n^f \times n^c$.

Set $P^{(tentative)} := 0$.

For each $j \in \{1, \dots, n^c\}$

{

- Define a local F -point subset, Ω_j , with $\Gamma_j \setminus \{i_j\} \subseteq \Omega_j \subseteq F$.
- Define a vector $z^{(local)}$ of size n^f .
- Set $z_i^{(local)} := \begin{cases} 1 & i = i_j \\ 0 & \text{otherwise} \end{cases}$.
- Perform $\nu_{(local)}$ relaxation sweeps on $z^{(local)}$ in Ω_j .
- Set $P_{i,j}^{(tentative)} := z_i^{(local)}$, $i \in \Gamma_j$.

}

Normalization

- Define a vector $z^{(global)}$ of size n^f .
- Set $z_i^{(global)} := 1$, $i = 1, \dots, n^f$.

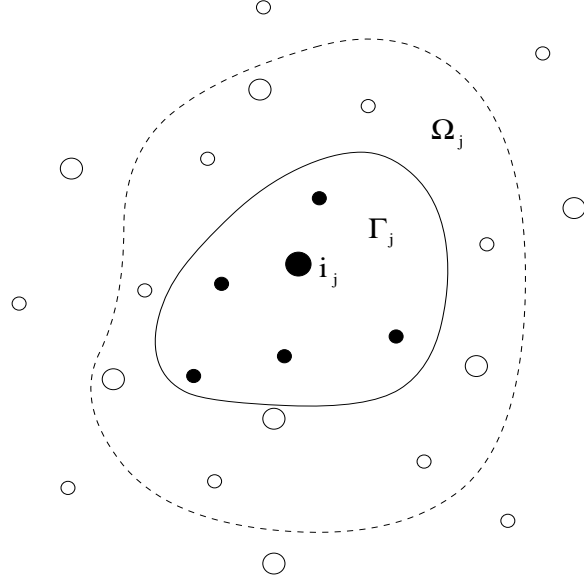


Figure 1: The first stage of Procedure 1 is illustrated. Large and small circles denote C points and F points, respectively. The large black circle denotes the C point i_j . Γ_j is shown by the set of black circles enclosed by the solid line. Ω_j is depicted by the set of small circles (black and white) enclosed by the dashed line. In the first stage of Procedure 1, all values $z_i^{(local)}$ are set to zero, except $z_{i_j}^{(local)}$ which is set to 1. Then, $\nu_{(local)}$ relaxation sweeps are performed in Ω_j . $P_{i,j}^{(tentative)}$ is given the values of $z_i^{(local)}$ thus obtained for $i \in \Gamma_j$. In particular, $P_{i_j,j} = 1$, since i_j is not included in Ω_j (being a C point), so it is not relaxed.

- Perform $\nu_{(global)}$ global F relaxation sweeps.
- Set $P_{i,j} := z_i^{(global)} \frac{P_{i,j}^{(tentative)}}{\sum_{1 \leq k \leq n^c} |P_{i,k}^{(tentative)}|}$, $i = 1, \dots, n^f$, $j = 1, \dots, n^c$.

The first stage of Procedure 1 is an approximate computation of the local influence of values defined at coarse-grid points, using homogeneous Dirichlet boundary conditions on $\partial\Omega_j$. This choice is cheap and simple, but it tends to underestimate the proper prolongation coefficients because the effects of all but a few nearby coarse-grid points are ignored, and also partly due to the fact that the local problem is solved only approximately by a few relaxation sweeps. In particular, if the row-sum of the i th row of A is zero, then the row-sum of the i th row of P should be one in order to obtain proper interpolation of the constant function. This is not generally achieved in the first stage, but the second stage (which is motivated by the discussion of prolongation strategies by Stüben in [20, Appendix A]) normalizes the prolongation such that a constant is interpolated correctly. In particular, note that if the sum of the i th row of A^f is zero, then the sum of the i th row of P is one, provided that $P_{i,j}^{(tentative)}$ are all nonnegative.

Amongst existing AMG algorithms, our approach is closest to the “element free AMGe” method of [12]. The main difference (in addition to some structural variations and the fact that we apply only a very small number of relaxation sweeps in Ω_j rather than attempt to solve the problem exactly) is that we do not try to determine appropriate local boundary conditions for the first stage of the procedure. Instead, we use homogeneous Dirichlet conditions and compensate for this with the normalization stage. This results in a cheap and simple procedure, as indicated below. This approach assumes (as do the classical AMG procedures) that the constant function needs to be interpolated well. Of course, other functions can be chosen as well if they are known to be important. We believe that the vectors that need to be interpolated (nearly) exactly must be given or determined by a separate (non-local) process. An approach for performing this task is currently being developed.

2.1.2 Application to the Structured Problem.

Procedure 1 describes a general approach which can be applied to unstructured problems. In the present problem our domain is a rectangular array of meshpoints, corresponding to the pixels of an image. Of course, we wish to take advantage of this structured mesh. Hence, we employ the “standard” coarsening for our partition (4). That is, the vector \bar{z} is considered as a two-dimensional array,

$$\bar{z} = \{z_{i,j}\}, \quad (i, j) \in [0, 1, 2, \dots, n_x - 1] \times [0, 1, 2, \dots, n_y - 1],$$

with a similar arrangement on all the grids. Point (i, j) is designated as a C point if both i and j are even, hence $n^c/n^f \approx 1/4$. Thus, the eight nearest neighbors of each C point, j , are F points (with the possible exception of points near the boundary). We choose Ω_j to be these eight F points, and $\Gamma_j = \{i_j\} \cup \Omega_j$. The rationale behind this choice is that the PDE is a second-order elliptic equation, and therefore a second-order prolongation (e.g., bilinear interpolation in the constant-coefficient case) is a good choice. Using only nearest neighbors would not be appropriate because, due to the five-point fine-grid stencil, about 1/4 of the fine-grid points have no coarse-grid nearest neighbors. On the other hand, including all neighbors of neighbors would lead to unnecessarily large prolongation stencils, resulting in correspondingly large coarse-grid operators.

Based on numerical tests, some of which are described in the next subsection, we employ Black-Red relaxation in both stages of the algorithm, with $\nu_{(local)} = \nu_{(global)} = 1$. This relaxation is composed of a “Black” step, whereby a Jacobi relaxation sweep is carried out at all meshpoints (i, j) with $i + j$ odd, followed by a “Red” relaxation step at the even-indexed points.

Let us estimate the computational cost of constructing the prolongation. Note that the four nearest neighbors of C points are “Black”, while the four diagonal near neighbors are “Red”. We assume here nine-point stencils. The local relaxation at the Black points requires only a single arithmetic operation per point (because the values at all the neighboring points are zero, except the C point where the value is one), and the subsequent relaxation of the Red points requires five operations per point. This adds up to 24 operations per C point, or about $6n^f$ operations. In comparison, we estimate that an exact solve can be performed with about 80 operations per C point (due to the special nearly-tridiagonal form of the local matrices), or about $20n^f$ operations. The single Black-Red global F relaxation in the second stage of the algorithm requires 8 operations for the Black points (of which there are $n^f/2$), and 12 for the Red points (of which there are $n^f/4$), for a total of $7n^f$ operations. The normalization in the final step of Procedure 1 costs about $3n^f$ arithmetic operations, excluding computation of absolute values. This adds up to $16n^f$ operations, which is less than the cost of a single ordinary full Red-Black fine-grid relaxation sweep. (This needs to be repeated for each of the coarse grids as well during the recursion, so the total number of operations comes to about $20n^f$.) Of course, this work can easily be carried out in parallel. The procedure is thus only a small part of the work required for defining the multigrid components, most of which is spent on the sparse-matrix multiplications required for computing the Galerkin coarse-grid operators, which is standard for robust multigrid methods.

A detailed discussion of how the parameters, in particular Γ_j and Ω_j , should be chosen in general unstructured settings is outside the scope of this paper and will be discussed separately. Here we only point out several observations. Obviously, these choices depend on the strategy for selecting C points. Clearly, Γ_j must include nearest neighbors of the C point i_j , and also, every F point must belong to at least one Γ_j . This would suffice if C points are chosen as in classical AMG, but we would like to allow a sparser C set. We might therefore also include neighbors of neighbors, but we should then “prune” the prolongation stencil by eliminating small values in order to avoid an unnecessary and costly increase in the size of the stencil. As for Ω_j , using $\Omega_j = \Gamma_j \setminus \{i_j\}$ seems sufficient, at least for compact stencils.

2.2 Numerical Test of Robustness

To test the reliability and efficiency of the solver, we discretize (2) by the five-point finite-difference discretization, impose constraints at certain points, and solve the resulting system with multigrid V -cycles, employing on each grid but the coarsest one relaxation sweep before transferring to the next-coarser grid and one sweep after obtaining the coarse-grid correction—a so-called $V(1, 1)$ cycle. We employ Red-Black relaxation, applying a Jacobi relaxation sweep at all points (i, j) with $i + j$ even, followed by a similar sweep over the remaining points. Of course, these are full sweeps, not F sweeps; the latter are employed only in the construction of the prolongation.

The solution is constrained at c random points, with $c = 1, 10, 100, 1000$, imposing random values. These constraints represent holes in the domain, where the solution is imposed and the differential equation is not satisfied. The weight function is given by $w(x, y) = r_{\min}^{-2}$, where r_{\min} is the

c	1	10	100	1000
64^2	0.171–0.214	0.137–0.187	0.097–0.139	0.048–0.079
128^2	0.111–0.220	0.161–0.189	0.134–0.163	0.096–0.137
256^2	0.124–0.216	0.166–0.214	0.157–0.176	0.119–0.129

Table 1: Asymptotic residual reduction factors per cycle, averaged over the last five cycles, are tabulated for resolutions 64 by 64, 128 by 128 and 256 by 256. c denotes the number of constrained points. $V(1, 1)$ cycles with Red-Black relaxation are employed. The best and the worst values obtained in ten test runs for each case are given.

distance to the nearest constrained point. (Note that w is only required midway between gridpoints, so it is finite everywhere we need to define it.) A first approximation to the solution is obtained by a full-multigrid (FMG) algorithm (see, e.g., [5, 20]), which is essentially a pyramidal outer loop with inner multigrid V cycles. Then, we apply ten $V(1, 1)$ cycles and compute the residual-norm reduction factor per cycle, geometrically averaged over the last five cycles. The values of p and q are obtained from a random image, but tests with $p = q = 0$ (i.e., Laplace equation) give very similar results. In fact, the asymptotic convergence factor is essentially independent of these data. More generally, an iterative process applied to a linear nonsingular problem $Ax = f^{(1)}$, with initial guess $x_0^{(1)}$, displays the same convergence history (up to roundoff errors) as the same process applied to the problem $Ax = f^{(2)}$ with a different initial guess, $x_0^{(2)} = x_0^{(1)} + A^{-1}(f^{(2)} - f^{(1)})$, because the initial residuals (and errors) are the same. Thus, the data do not affect the asymptotic convergence rate, because this rate is generally independent of the initial guess (since the modes that converge most slowly “creep in” via roundoff effects even if they are somehow excluded initially.) However, the convergence behavior obviously does depend on the location and number of constraints, because these affect A . Hence, each test is run ten times—for ten different random locations of the constrained points—and the best (smallest) and worst (largest) residual reduction factors per cycle are shown in Table 2.2. The tests are carried out at resolutions 64 by 64, 128 by 128 and 256 by 256. We find that the method is robust for this set of problems. The convergence behavior actually improves when the number of constraints is increased, and it is not sensitive to the random location of the constrained points. In the worst case, the error is still reduced by a factor of more than 4 per cycle, so two cycles should easily suffice to obtain accuracy that is comparable to the discretization error if the FMG algorithm is employed [20, §3.2.2]. Thus, the goal of solution in $O(n)$ operations is met for this set of problems. There is no significant deterioration as the resolution is increased. (The slight advantage of the lower-resolution results when c is large is due, at least partly, to the larger c/n ratio, which results in a greater “average” diagonal dominance; indeed, when c/n is appreciable, the relaxation process itself is found to have good convergence properties.) We also performed a similar set of tests to determine the optimal value for $\nu_{(global)}$. We found $\nu_{(global)} = 1$ to be best (i.e., achieve the best convergence rate) in *all* our tests. Similar tests using Jacobi local and global F relaxation instead of Black-Red in the prolongation construction showed comparable behavior with $\nu_{(local)} = \nu_{(global)} = 2$.

The numerical tests were repeated, for resolution 128^2 , using an exact solve instead of $\nu_{(local)} = 1$. The asymptotic convergence rates in these tests were better (smaller) than the ones shown above, typically by just two or three percent. The worst behavior in this respect was exhibited for

$c = 1$, where the average deterioration (over ten runs) was by 4.4%. This implies that the expected residual reduction by 100 cycles using $\nu_{(local)} = 1$ is better than that of 95 cycles using exact solves. Since only one or two cycles actually need to be used in practice, this difference is negligible. The computational savings due to the approximate solution are implementation and problem dependent. In general, for each coarse-grid point one needs to construct a linear system of $|\Omega_j|$ equations and variables, and the difference is between solving this system exactly or performing just a small number of relaxation sweeps. In the particular case examined here, we estimate above that the exact solve is about 3.3 times as expensive as a single relaxation, using the best implementation. Of course the overall savings are less significant than this, because this is just a part of the prolongation construction. In more general settings, the cost of obtaining an exact solution quickly grows with $|\Omega_j|$, so if the number of required relaxation sweeps remains small, significant savings may be obtained. The reason why the required number of relaxation sweeps should remain small is the strong diagonal dominance of the local systems (due to the fact that coefficients multiplying C -points have been eliminated).

3 Experimental Results

3.1 Shape from photometric stereo

In our first example we compute shape from photometric stereo with and without constraints. (Actually, the unconstrained problem does require a single arbitrary constraint, because the solution is otherwise only determined up to a freely chosen translation due to the Neumann boundary conditions.) Fig. 2 is a set of five input images from which we reconstruct the surface. (We use five images with averaging, rather than only three, to compensate somewhat for the shadow and specular effects.) We test our method with and without a small number of constraints. To obtain



Figure 2: *The input images for the shape from stereo reconstruction process. Same camera position and head object with different lighting directions.*

the constraints, the relative height values at eight points were manually extracted by matching corresponding points in the two views of the object shown in Fig. 3 (Top). Next, the reconstruction algorithm was applied to the images with and without the constraints. As shown in Fig. 4, the reconstruction with the given data points better captures the 3D structure of the surface, while gracefully interpolating between the eight given points without generating any artificial discontinuities.

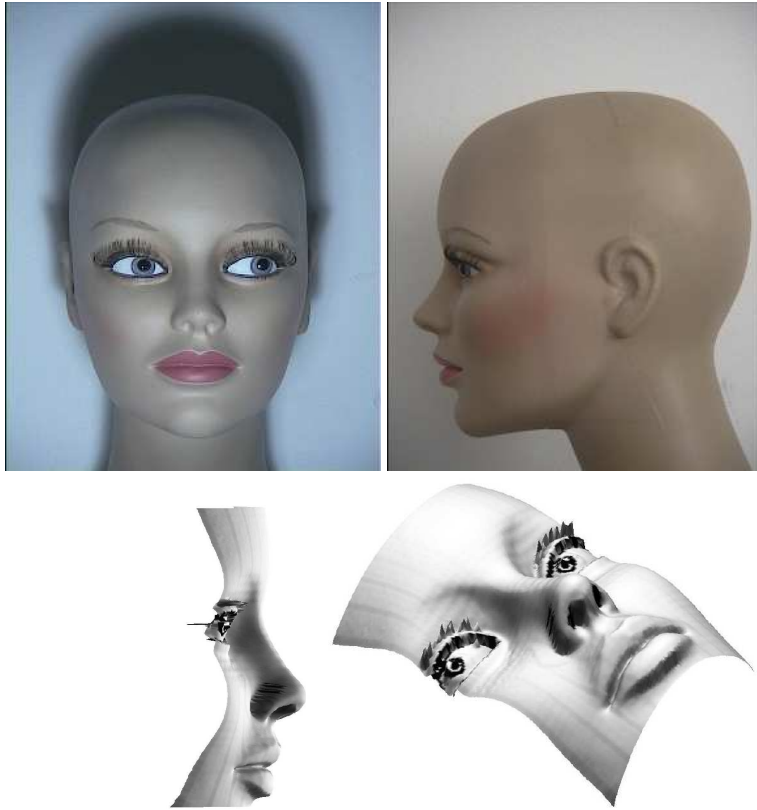


Figure 3: *Top: side and frontal views from which height values at a small number of points were extracted to improve the reconstruction. Bottom: three-dimensional measured height-map, obtained with a Cyberware Laser Scanner.*

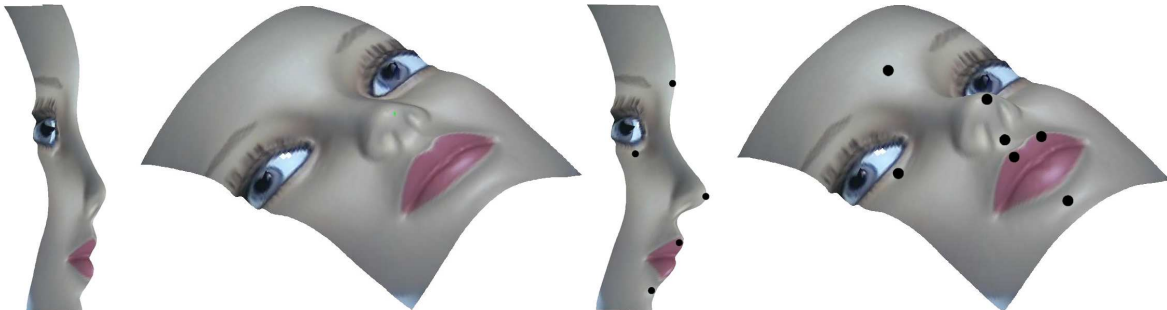


Figure 4: *Side and perspective views of the reconstructed surface, with the frontal texture mapped onto it. Left frames: with no constraint. Right frames: with eight (marked) constrained points.*

3.2 The Yanowitz-Bruckstein binarization method

The Yanowitz-Bruckstein method [22] is a technique for image binarization which handles non-uniformly illuminated images. See [18, 19] for a comparison and an overview of binarization methods for text analysis. The algorithm is based on the simple observation that a proper threshold surface (defined below) may be defined by the surface interpolating image points located at the centers of edges, where edges are defined at the locations where, for example, the image has high gradient magnitude.

In the Yanowitz-Bruckstein binarization method, first the locations of the edge centers are detected. The image values at these points serve as constraints for a Laplace equation whose solution serves as a threshold surface. The binary image is then obtained by thresholding the original image. That is, given a gray level image $I(x, y)$, the binary image gets a value 1 if the image gray level at a point is higher than the value of the computed threshold surface $u(x, y)$, and gets a zero value otherwise.

The main difficulty with the Yanowitz-Bruckstein method is that classical SOR relaxation methods for solving the Laplace equation with sparse constraints at the edge locations require many iterations to converge, especially if the constrained points are far from one another. The AMG solver again provides an excellent remedy for this problem.

More formally, given the image $I(x, y) : \Omega \rightarrow \mathbb{R}^+$, the algorithm for computing the threshold surface $u(x, y)$ proceeds as follows:

- Isolate the locations of edge centers, for example by the set of points, $e = \{(x, y) : |\nabla I| > T\}$, for some given threshold T .
- Use the values $I(x, y)$ as constraints at the set of points $(x, y) \in e$ to solve for the threshold surface $\Delta u = 0$ in $\Omega \setminus e$, subject to $u(x, y) = I(x, y), \forall (x, y) \in e$.

The surface u is the minimizer of $\int |\nabla u|^2 dx dy$ that is constrained to pass through the points $I(x, y)$ for $(x, y) \in e$. Again, one may smooth the interpolation surface near the edge points by adding a weighting function $w(x, y)$ that gets higher values near the points in e . The minimization functional is slightly simpler than in the previous example and reads $\int w |\nabla u|^2 dx dy$. We can use our shape from photometric stereo algorithm by setting p and q to zero.

We show two examples of a 256×256 map image, to which we first add a tilted intensity plane defined by the gray level image $n(x, y) = x + y$, and then the same image added to the intensity image $n(x, y) = x^2 + y^2$, centered in the middle of the image. See Fig. 5 for the input images, and Fig. 6 for the binarization results.

4 Conclusions

A new algebraic multigrid numerical method was applied to two classical problems of image analysis. The Horowitz-Kiryati problem [13] of shape from photometric stereo with constraints was solved efficiently and robustly by the AMG algorithm. The reconstructed surface satisfies a variational principle without compromising the physical model, while also satisfying a discontinuous set of constraints by passing smoothly through given points and curves on the surface. In the second example the AMG method was applied to the image binarization problem, as the main step in the Yanowitz-Bruckstein binarization procedure.

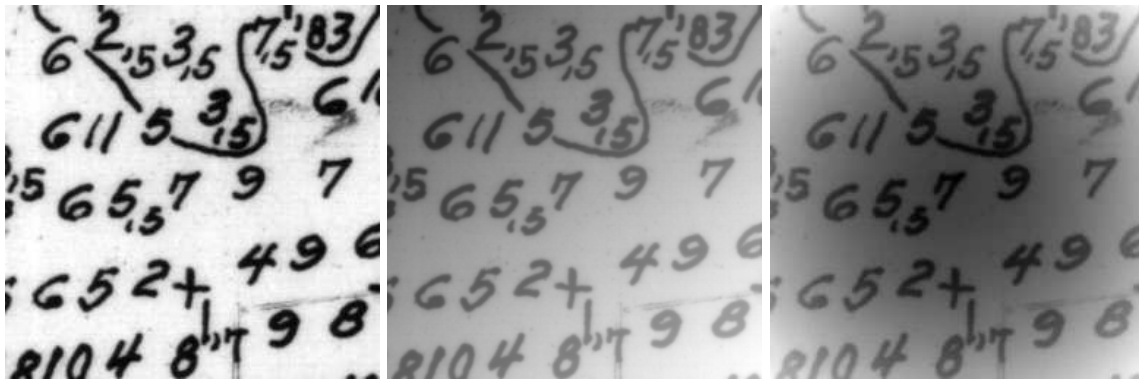


Figure 5: Left: Original map image. Middle: Adding a tilted intensity plane. Right: Adding a $x^2 + y^2$ intensity surface.

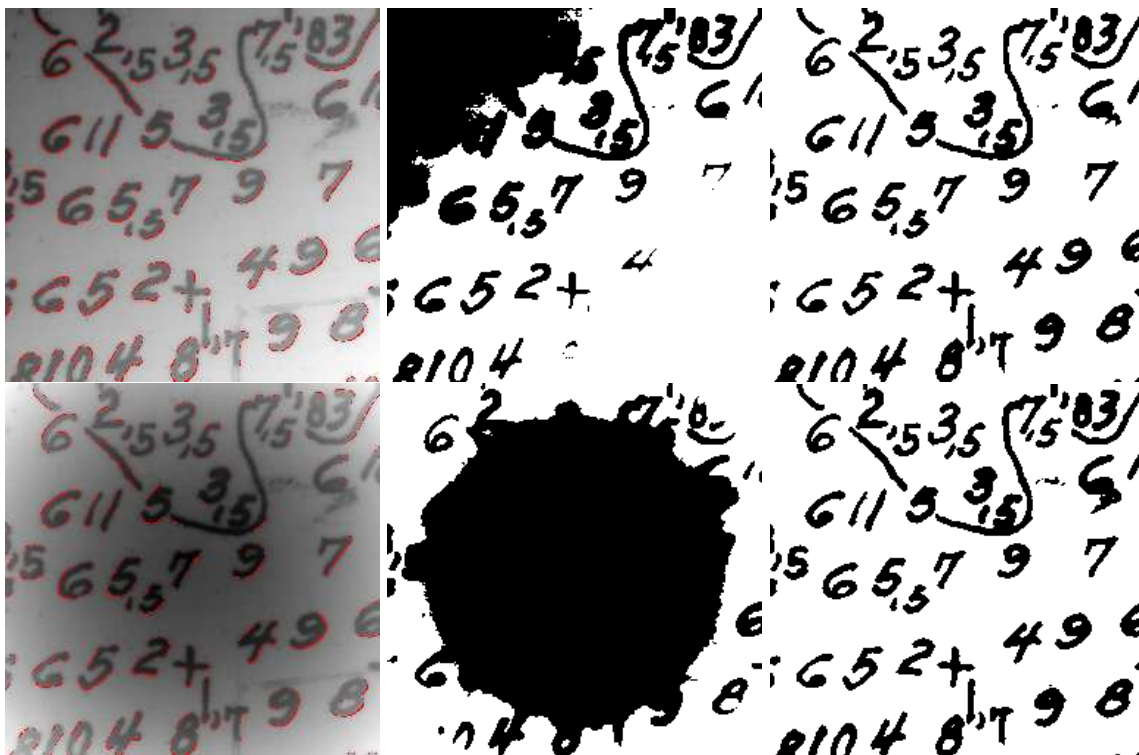


Figure 6: Left: Input image with edge locations marked red. Middle: Naive thresholded binary image $I - \max(I)/2$. Right: Yanowitz-Bruckstein adaptive thresholded binary image $I - u$.

The AMG algorithm was tested on a set of problems with randomly distributed pointwise constraints and singular weight functions, and was found to perform robustly and efficiently. The proposed prolongation construction is computationally inexpensive, and the setup phase is correspondingly efficient. We conclude that the proposed multigrid method performs essentially optimally for these problems. Simpler methods may be faster in such special situations where the number of constraints is very small (e.g., classical multigrid with Krylov subspace acceleration) or very large (simple relaxation), but the AMG approach is robust for the full range of problems, and it is therefore the method of choice. Generalization of the method to unstructured problems is under way and will be reported elsewhere.

The shape from photometric stereo and the binarization method are just two examples in which elliptic boundary-value problems with irregular constraints are encountered in image processing and computer vision. Further examples are shape from shading, lightness, and optical flow [7, 16, 17], all of which can be coupled efficiently with irregular constraints using the multigrid solver. Another example is efficient image reconstruction from data around the edges in so called ‘second generation’ geometric based image coding [6, 10].

Acknowledgments

The advice of Nahum Kiryati of the Tel-Aviv University is gratefully acknowledged. The AMG method was developed by the second author in collaboration with Achi Brandt of the Weizmann Institute of Science, while the author was visiting the Institute of Geophysics and Planetary Physics in the University of California at Los Angeles. All calculations were performed at the Technion. We thank Eyal Gordon of the Geometric Processing Group at the Technion for his help in producing Fig. 3.

References

- [1] R. E. Alcouffe, A. Brandt, J. E. Dendy, and J. W. Painter. The multi-grid method for the diffusion equation with strongly discontinuous coefficients. *SIAM J. Sci. Stat. Comput.*, 2:430–454, 1981.
- [2] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. Comp.*, 31:333–390, 1977.
- [3] A. Brandt. General highly accurate algebraic coarsening schemes. *Electronic Trans. Numer. Anal.*, 10:1–20, 2000.
- [4] A. Brandt, S. F. McCormick, and J. Ruge. Algebraic multigrid (AMG) for sparse matrix equations. In D. J. Evans, editor, *Sparsity and its applications*, pages 257–284. Cambridge University Press, Cambridge, 1984.
- [5] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A multigrid tutorial*. SIAM, Philadelphia, second edition, 2000.
- [6] S. Carlsson. Sketch based coding of grey level images. *Signal Processing*, 15:57–83, 1988.

- [7] A. K. Chhabra and T. Grogan. On Poisson solvers and semi-direct methods for computing area based optical flow. *IEEE Trans. on PAMI*, 16:1133–1138, 1994.
- [8] P. M. de Zeeuw. Matrix-dependent prolongations and restrictions in a blackbox multigrid solver. *J. Comput. Appl. Math.*, 33:1–27, 1990.
- [9] J. E. Dendy (Jr.). Black box multigrid. *J. Comput. Phys.*, 48:366–386, 1982.
- [10] J. Froment and S. Mallat. Second generation compact image coding with wavelets. In C. Chui, editor, *Wavelets: A Tutorial in Theory and Applications*, pages 655–678. Academic Press, 1992.
- [11] S. Henn and K. Witsch. A multigrid approach for minimizing a nonlinear functional for digital image matching. *Computing*, 64:339–348, 2000.
- [12] V. E. Henson and P. S. Vassilevski. Element-free AMGe: general algorithms for computing interpolation weights in AMG. *SIAM J. Sci. Comput.*, 23(2):629–650, 2001.
- [13] I. Horowitz and N. Kiryati. Bias correction in photometric stereo using control points. Technical Report, Department of Electrical Engineering—Systems, Tel Aviv University, Israel, 2000.
- [14] T. F. C. R. H. Chan and W. L. Wan. Multigrid for differential-convolution problems arising from image processing. CAM report 97-20, UCLA, 1997.
- [15] J. Ruge and K. Stüben. Algebraic multigrid (AMG). In S. F. McCormick, editor, *Multigrid methods, frontiers in applied mathematics*, pages 73–130. SIAM, Philadelphia, 1987.
- [16] T. Simchony, R. Chellappa, and M. Shao. Direct analytical methods for solving Poisson equations in computer vision problems. *IEEE Trans. on PAMI*, 12(5):435–446, 1990.
- [17] D. Terzopoulos. Image analysis using multigrid relaxation methods. *IEEE Trans. on PAMI*, 8(2):129–139, 1986.
- [18] O. D. Trier and A. K. Jain. Goal-directed evaluation of binarization methods. *IEEE Trans. on PAMI*, 17(12):1191–1201, 1995.
- [19] O. D. Trier and T. Taxt. Evaluation of binarization methods for document images. *IEEE Trans. on PAMI*, 17(3):312–315, 1995.
- [20] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, London and San Diego, 2001.
- [21] M. Unser. Multigrid adaptive image processing. In *Proc. IEEE International Conference on Image Processing (ICIP)*, 1995.
- [22] S. D. Yanowitz and A. M. Bruckstein. A new method for image segmentation. *Computer Vision, Graphics, and Image Processing*, 46:82–95, 1989.