An $O(N)$ Iterative Solution to the Poisson Equation in Low-level Vision Problems[1]

S. H. Lai and B. C. Vemuri

UF-CIS Technical Report TR-93-035

Computer and Information Sciences Department

University of Florida

Bldg. CSE, Room 301

Gainesville, Florida 32611

November 15, 1993

[1]This work was supported in part by the NSF grant ECS-9210648.

In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, Washington 1994.

In this paper, we present a novel iterative numerical solution to the Poisson equation whose solution is needed in a variety of low-level vision problems. Our algorithm is an $O(N)$ (N being the number of discretization points) iterative technique and does not make any assumptions on the shape of the input domain unlike the polyhedral domain assumption in the proof of convergence of multi-grid techniques [28, 29]. We present two major results namely, a generalized version of the capacitance matrix theorem [6] and a theorem on $O(N)$ convergence of the alternating direction implicit method (ADI) used in our algorithm. Using this generalized theorem, we express the linear system corresponding to the discretized Poisson equation as a Lyapunov and a Capacitance matrix equation. The former is solved using the ADI method while the solution to the later is obtained using a modified bi-conjugate gradient algorithm. We demonstrate the algorithm performance on synthesized data for the surface reconstruction and the SFS problems.

1 Introduction

The problem of reconstructing and representing three-dimensional shapes has received an enormous amount of attention in vision research for the past decade. In this paper we will be concerned with the problem of surface shape recovery from shading and sparse depth information. These problems may be formulated in the framework of variational principles which lead to solving Euler-Lagrange equations as the necessary condition for a minimum. In these problems, there is a need to solve one or more Poisson equations of the form $\Delta v = f$. Several other problems in low-level vision can be formulated in the variational principle framework and lead to solving one or more discrete Poisson equations. Our selection of the surface reconstruction and shape from shading problem in this paper has no predisposed preference. The Poisson equation (an elliptic PDE), when discretized leads to a large sparse linear system which may be solved by using either direct or iterative numerical methods.

In the iterative solution category, we have numerous techniques namely, Gauss-Seidel [7], successive over relaxation (SOR), conjugate gradient [9], hierarchical conjugate gradient (HCG) [20], and multigrid methods. The slowest being Gauss-Seidel, taking $O(N^2)$ time to converge for an $N \times N$ problem [9] while, the fastest being multi-grid methods taking $O(N)$ time to converge [10]. Although the multi-grid techniques have been applied successfully to a general class of problems,

the proofs of convergence are *restricted* to a special class of problems [10]. Hackbusch [10] presents a general concept for proving the convergence of multi-grid iterations under some assumptions on the regularity of the domain boundary. Braess [5] proved the convergence of the multi-grid solution to the Poisson equation on a uniform grid and showed that the results are independent of the shape of the domain as long it is polygonal and convex. Recently, Xu [28] presents a unified theory for classifying a class of iterative solution methods for symmetric positive definite problems. An abstract convergence theory is established and applied to a particular problem by specifying, a decomposition of the underlying space, and the corresponding subspace solvers. With multi-grid methods, the subspaces are given by multiple coarser grids. As a direct consequence of the abstract theory, optimal convergence estimates for a given algorithm can be obtained via the estimation of only two parameters (see Xu [28] for details). Although, results for convergence of multi-grid methods for complicated domains have been discussed in [28, 29], the analysis is limited to domains with Lipschutz boundaries and has to be done on a case by case basis and is different for each case.

Direct solutions to the discretized Poisson equation (for irregular regions) are based on a theorem called the capacitance matrix theorem which was stated and proved in Buzbee *et al.*, [6]. The direct solutions using the capacitance matrix theorem employ the Fourier-Toeplitz method in combination with LU decomposition [19], the Cholesky decomposition [6] or the conjugate gradient [18] technique. The Fourier-Toeplitz method requires $O(N \log N)$ time and the LU decomposition, Cholesky factorization or the conjugate gradient require $O(n^3) = O(N\sqrt{N})$. Thus, making the overall complexity $O(N\sqrt{N})$.

In this paper, we present an iterative $O(N)$ solution to the Poisson equation, which does not make any assumptions on the domain shape or its regularity and treats them all in a uniform manner. Our algorithm is based on the generalized version of the capacitance matrix theorem which we state and prove in this paper. In Our algorithm we first transform the linear system of equations into a Lyapunov matrix equation and an associated linear system with a capacitance matrix. The former is solved using the alternating direction implicit method (ADI) while the solution to the later is obtained using a modified bi-conjugate gradient technique. The total time complexity of our algorithm is the sum of the time complexities of the ADI and the bi-conjugate gradient methods. We prove that the ADI method takes a constant number of iterations with each iteration taking

$O(N)$ time. In our proof, we make use of the spectral characteristic of the Laplacian operator which appears on the left hand side of the Poisson equation and impose a very mild restriction on the spectral characteristics of the input data. This restriction amounts to requiring that the frequency content of the data be upper bound by an increasing function of $\|\omega\|_2$ (ω is the frequency domain vector valued variable) of degree less than 1. In all low-level vision problems considered in this paper and many others, this restriction is not violated. For the bi-conjugate gradient algorithm, which requires a (dense) matrix vector multiplication in each iteration, we propose to carry out this operation in a wavelet basis. Approximation to matrix vector multiplication in a wavelet basis requires only $O(\sqrt{N}) = O(n)$ time. Thus, the bi-conjugate gradient algorithm with the aforementioned wavelet basis approximation to the matrix vector products will require $O(N)$ time since there are $O(\sqrt{N}) = O(n)$ iterations in the algorithm. Hence, the total time complexity of our algorithm is $O(N)$. We will present the details of our algorithm in subsequent sections.

The rest of the paper is organized as follows: in section 2, we present various low-level vision problems whose formulations lead to solving the discretized Poisson equation, section 3 contains our formulation for the solution to the Poisson equation and a "flow chart" of our algorithm along with a description of the individual blocks of the "flow chart" are presented in section 3.3. In section 4 we present a numerical solution to the discretized Poisson equation and prove the convergence of the ADI iterative technique applied to our problem. A modified bi-conjugate gradient algorithm for solving the capacitance matrix equation is also presented. In section 5, we present examples of our algorithm applied to synthetic data for the surface reconstruction and the shape from shading problems and finally conclude in section 6.

2 Low-level Vision Problems and Previous Work

There are many problems in low-level vision which when formulated as variational principles lead to solving one or more discretized Poisson equations. Some of these problems are: the surface reconstruction [3, 23, 26, 4] shape from shading [12, 19], lightness [11] and optical flow [13, 19]. In this paper, we will very briefly discuss the formulation of the first two of these problems as variational principles and without dwelling much on them, will point out the structure of the matrices that

appear in the linear system which needs to be solved for surface shape recovery.

2.1 Variational Principle Formulation

Variational formulations for various low-level vision problems have been reported in Poggio et al., [17]. These formulations make use of the popular theory of regularization. In a regularization framework, generic smoothness assumptions are imposed on the solution space prior to attempting any functional minimization. The smoothness constraints are well characterized by a class of generalized multi-dimensional spline functionals [22]. The formulations involve minimization of an energy functional $\mathcal{E}$, which is the sum of the energy contribution from the smoothness constraint ($\mathcal{E}_s$) and the data constraint ($\mathcal{E}_d$) i.e.,

$$Find \;\; u \;\; such \;\; that \; \mathcal{E}(u) = \overset{inf}{v \in \mathcal{H}} \;\; \mathcal{E}(v) \tag{1}$$

where, $\mathcal{H}$ defines the linear admissible space of smooth functions defined on $\Re^2$ and the functional

$$\mathcal{E}(v) = \mathcal{P}(v) + \lambda \mathcal{S}(v) \tag{2}$$

Where, λ is called the regularization parameter that controls the contribution of the smoothness term, $\mathcal{S}(v) : \mathcal{H} \mapsto \Re$ is a functional on $\mathcal{H}$ that is a measure of smoothness of an admissible function $v(x, y)$ and $\mathcal{P}(v) : \mathcal{H} \mapsto \Re$ is a functional that measures the discrepancy between the function and the given data [22].

We will now proceed to discuss the specific functionals used for $\mathcal{S}(v)$ and $\mathcal{P}(v)$ in the surface reconstruction and the SFS problems.

2.1.1 Surface Reconstruction

Surface reconstruction from range data has been intensely studied for the past decade by many researchers in the computer vision community [3, 23, 20, 26, 4, 25, 24]. Variational Splines [22] have emerged as the single most popular solution to the surface reconstruction problem. In the following, we give the precise expression for the smoothness and data constraints used in the variational

formulation of surface reconstruction problem. The smoothness constraint using only the lowest order derivative term can be written as,

$$\mathcal{S}(v) = \int\int_{\Omega} (v_x^2 + v_y^2)\} \, \mathrm{d}x\mathrm{d}y, \tag{3}$$

where $v(x, y)$ is the admissible function and v_x, v_y its partial derivatives assumed to be small.

The above energy expression can be interpreted as the deflection energy in a membrane (e.g., a rubber sheet) and serves as a stabilizer in the overall variational principle for the surface reconstruction problem. To the stabilizer, we add data constraints via what are known as penalty terms. The following penalty term which measures the discrepancy between the surface and data weighted by the uncertainty in the data may be used,

$$\mathcal{P}(v) = \frac{1}{2} \sum c_i \left(v(x_i, y_i) - d_i\right)^2 \tag{4}$$

Where, d_i are the depth data points specified in the domain Ω and c_i are the uncertainty associated with the data. The total energy is $\mathcal{E} = \lambda \mathcal{S}(v) + \mathcal{P}(v)$. Where, λ is the regularization parameter that controls the amount of smoothing performed. The goal is to find a u that minimizes the total potential energy $\mathcal{E}(v)$.

To compute a numerical solution to the above minimization problem, we first discretized the functionals $\mathcal{S}(v)$ and $\mathcal{P}(v)$ using finite element techniques [23]. The energy due to the data compatibility term in discrete form becomes,

$$E_d(\mathbf{x}, \mathbf{d}) = \frac{1}{2}(\mathbf{x} - \mathbf{d})^T \mathbf{K}_d (\mathbf{x} - \mathbf{d}). \tag{5}$$

Where $\mathbf{x}$ is the discretized surface, $\mathbf{d}$ are the data points, and $\mathbf{K}_d$ is a diagonal matrix (for uncorrelated noise in the data) containing the uncertainty σ_i associated with the data points. The smoothness energy in discrete form is

$$E_s(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{K}_s \mathbf{x} \tag{6}$$

where $\mathbf{K}_s$ is a very large $(n^2 \times n^2)$, sparse and banded matrix called the stiffness matrix. The resulting energy function is a quadratic in $\mathbf{x}$ given by,

$$E(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{K} \mathbf{x} - \mathbf{x}^T \mathbf{b} + c \tag{7}$$

with $\mathbf{K} = \lambda \mathbf{K}_s + \mathbf{K}_d$ and $\mathbf{b} = \mathbf{K}_d \mathbf{d}$. The minimum of this energy function $\mathbf{u}$ is found by solving the large sparse linear system $\mathbf{K}\mathbf{x} = \mathbf{b}$. The structure of the stiffness matrix $\mathbf{K}$ can be analyzed using the computational molecules [23]. We present our fast solution to this problem in a subsequent section.

2.1.2 Shape from Shading

Like the surface reconstruction problem, the shape from shading problem has received enormous amount of attention in the past decade. Numerous solution methods using variational principle formulations have been proposed. For a comprehensive set of papers on this topic, we refer the reader to the book edited by Horn and Brooks [12] and the work in [15, 21, 19].

In this problem, it is required to recover the shape of surfaces from image irradiance which depends on surface geometry and reflectance, scene illuminance and imaging geometry. The image irradiance can be expressed directly as a function of the surface orientation if illuminance, reflectance and imaging geometry are assumed to be constant. The shape from shading problem is commonly posed as a nonlinear, first order partial differential equation in two unknowns, called the image irradiance equation, $E(x, y) = R(p, q)$ [12]. Where, $E(x, y)$ is the image irradiance at a point (x, y), $p = v_x, q = v_y$ are first partials of the surface function $v(x, y)$, and $R(p, q)$ is the relation between surface orientation (p, q) and image irradiance $E(x, y)$. To overcome the ambiguity caused along occluding contours in the gradient space parameterization (p, q), a reparameterization of surface orientation in terms of a stereographic mapping: $f = 2ap,\ g = 2aq$ with $a = 1/(1+\sqrt{1+p^2+q^2})$ is used. With this reparameterization, the overall energy function (sum of stabilizer and data energies) to be minimized is expressed as

$$\mathcal{E}(f, g) = \lambda \int\int_{\Omega} (f_x^2 + f_y^2) + (g_x^2 + g_y^2) dx dy + \int\int_{\Omega} [E(x, y) - R(f, g)]^2 dx dy. \tag{8}$$

The Euler Lagrange equations which express the necessary condition for a minimum are given by the system of coupled PDEs

$$\left.\begin{array}{rcl} \Delta f &=& \lambda [R(f, g) - E(x, y)] R_f \\ \Delta g &=& \lambda [R(f, g) - E(x, y)] R_g \end{array}\right\} \tag{9}$$

The above equations may be discretized and solved using the algorithm of Simchony et al., [19] which enforces integrability condition on p and q. The method involves solving three different discretized Poisson equations namely, $\mathbf{K}\mathbf{x}_1 = \mathbf{b}_1$, $\mathbf{K}\mathbf{x}_2 = \mathbf{b}_2$, and $\mathbf{K}_1\mathbf{x}_3 = \mathbf{b}_3$. Where, $\mathbf{K}$ is the stiffness matrix obtained by discretizing the Laplacian operator (on the LHS of the equation 9) with Dirichlet boundary conditions. $\mathbf{K}_1$ is the stiffness matrix obtained by discretizing the Poisson equation for the depth from orientation problem i.e., $\Delta z = p_x + q_y$ with Neumann boundary conditions. $\mathbf{b}_1, \mathbf{b}_2$ are the appropriate RHSs obtained by discrete computation of the RHS of equation 9 at the current estimated value of (f, g). $\mathbf{b}_3$ is the discrete form of the RHS of depth from orientation equation for estimated (p, q) from equation 9. The structure of the stiffness matrices $\mathbf{K}$ can be analyzed using the computational molecules [23]. In the next section, we propose a fast solution to the above discretized Poisson equations.

3 Proposed Algorithm for Solving the Poisson Equation

In this section, we will propose a new algorithm to solve the algebraic system of equations resulting from the discretization of the Poisson equation or the minimization of the discrete version of the constrained variational principle discussed in the previous section. This algorithm is based on the generalized version of the capacitance matrix theorem that we state and prove in this section.

3.1 Capacitance Matrix Technique

The Capacitance matrix technique [6] has been used to solve discrete elliptic PDEs on irregular domains for the past several decades. This technique involves expressing the stiffness matrix $\mathbf{K}$ as the sum of $\mathbf{K}_0$ and $\mathbf{U}\mathbf{V}^T$. Where, we can choose $\mathbf{K}_0$ to be a well structured matrix that is slightly different from the original stiffness matrix $\mathbf{K}$ for the given PDE with certain boundary conditions and $\mathbf{U}\mathbf{V}^T$ encodes the difference between $\mathbf{K}$ and $\mathbf{K}_0$. Thus, the solution to the linear system $\mathbf{K}\mathbf{x} = \mathbf{b}$ can be obtained by solving the alternate linear system $\mathbf{K}_0\mathbf{x} = \mathbf{b}'$ and the associated linear system with a capacitance matrix which will be discussed subsequently. $O(N \log N)$ numerical methods may be used to solve the linear system $\mathbf{K}_0\mathbf{x} = \mathbf{b}'$, for example, the Fourier-Toeplitz method [19, 6]. *In this paper, we will present an $O(N)$ ADI method for solving the same.* As for the capacitance matrix equation, the solution depends on the dimensionality of $\mathbf{U}\mathbf{V}^T$ which is typically $O(\sqrt{N})$.

In the past, Buzbee et al., [6] have used the LU decomposition ($O(n^3) = O(N\sqrt{N})$ to solve the Capacitance matrix equation. We present a modified bi-conjugate gradient technique that takes $O(N)$ time for solving the same.

The Sherman-Morrison-Woodbury formula [9] can be applied to solve $(\mathbf{K_0} + \mathbf{UV}^T)\mathbf{x} = \mathbf{b}$ only when $\mathbf{K}$ and $\mathbf{K_0}$ are nonsingular. When the $rank(\mathbf{K_0}) = (N-1)$ and $\mathbf{K}$ is nonsingular, the capacitance matrix theorem as stated in Buzbee et al., [6] can be applied to obtain the solution. However, in many applications, $\mathbf{K}$ may be singular and $rank(\mathbf{K_0}) < (N-1)$ e.g., the Poisson equation with Neumann boundary conditions yields a singular $\mathbf{K}$ matrix; for higher order PDEs such as the biharmonic equation, the $rank(\mathbf{K_0}) \leq (N-3)$. These problems can be solved via the application of the generalized capacitance matrix theorem which is stated and proved below.

Theorem 1 *Given* $\mathbf{K} = \mathbf{K_0} + \mathbf{UV^T}$, *where* $\mathbf{K}, \mathbf{K_0} \in \Re^{N\times N}$, $\mathbf{U} = [\mathbf{u}_0, \ldots, \mathbf{u}_{p-1}]$, $\mathbf{V} = [\mathbf{v}_0, \ldots, \mathbf{v}_{p-1}]$, *with* $\mathbf{u}_0, \ldots, \mathbf{u}_{p-1}$ *being linearly independent vectors, and* $p = O(n)$ *where,* $n = \sqrt{N}$. *Assume* $rank(\mathbf{K_0}) = N - m$, *and let* $\mathbf{q}_i, i = 0, \ldots, m-1$, *be the eigenvectors of* $\mathbf{K}_0$ *corresponding to zero eigenvalues, i.e.* $\mathbf{K}_0\mathbf{q}_i = \mathbf{0}$, *such that* $\mathbf{q_i^T q_j} = \delta(i,j)$, $0 \leq i,j \leq m-1$, *and let* $\mathbf{q_i^T u_j} = \alpha_i\delta(i,j), for \;\; 0 \leq i,j \leq m-1$, *where the constant* $\alpha_i \neq 0$. *Let* $\bar{\mathbf{x}}$ *be a solution to*

$$\mathbf{K}_0\bar{\mathbf{x}} = \mathbf{b} - \sum_{j=0}^{m-1} \frac{\mathbf{q}_j^T\mathbf{b}}{\mathbf{q}_j^T\mathbf{u}_j}\mathbf{u}_j, \tag{10}$$

where $\mathbf{b} \in range(\mathbf{K})$. *For* $0 \leq i \leq p-1$, *let* $\boldsymbol{\eta}_i$ *be a solution to*

$$\mathbf{K}_0\boldsymbol{\eta}_i = \mathbf{u}_i - \sum_{j=0}^{m-1} \frac{\mathbf{q}_j^T\mathbf{u}_i}{\mathbf{q}_j^T\mathbf{u}_j}\mathbf{u}_j, \tag{11}$$

and $\boldsymbol{\eta}_0 \ldots \boldsymbol{\eta}_{m-1}$ *are nonzero vectors. Define the capacitance matrix* $\mathbf{C}$ *as*

$$\mathbf{C} = \mathbf{I} + \mathbf{V^T}\boldsymbol{\eta} - \sum_{j=0}^{m-1} \mathbf{e}_{j+1}\frac{\mathbf{q}_j^T\mathbf{U}}{\mathbf{q}_j^T\mathbf{u}_j}, \tag{12}$$

where $\boldsymbol{\eta} = [\boldsymbol{\eta}_0, \ldots, \boldsymbol{\eta}_{p-1}]$, *and* $\mathbf{e}_j \in \Re^{p\times 1}$ *is the unit vector with 1 at the j-th component and 0 elsewhere. Assume there is a solution* $\boldsymbol{\beta}$ *to*

$$\mathbf{C}\boldsymbol{\beta} = \mathbf{V}^T\bar{\mathbf{x}} - \sum_{j=0}^{m-1} \frac{\mathbf{q}_j^T\mathbf{b}}{\mathbf{q}_j^T\mathbf{u}_j}\mathbf{e}_j, \tag{13}$$

then $\bar{\mathbf{x}} - \boldsymbol{\eta}\boldsymbol{\beta}$ *is a solution to the linear system* $\mathbf{Kx} = \mathbf{b}$.

Proof: To prove this theorem, we will show that $\mathbf{K}(\bar{\mathbf{x}} - \boldsymbol{\eta}\boldsymbol{\beta}) = \mathbf{b}$. Expanding by substituting, $\mathbf{K}$ for $\mathbf{K_0} + \mathbf{UV^T}$, we have

$$\mathbf{K}(\bar{\mathbf{x}} - \boldsymbol{\eta}\boldsymbol{\beta}) = \mathbf{K}_0\bar{\mathbf{x}} - \mathbf{K}_0\boldsymbol{\eta}\boldsymbol{\beta} + \mathbf{U}\mathbf{V}^T\bar{\mathbf{x}} - \mathbf{U}\mathbf{V}^T\boldsymbol{\eta}\boldsymbol{\beta}. \tag{14}$$

In this equation, we can use 10 and 11 for $\mathbf{K}_0\bar{\mathbf{x}}$ and $\mathbf{K}_0\boldsymbol{\eta}$ respectively. For, $\mathbf{V}^T\boldsymbol{\eta}\boldsymbol{\beta}$, we use equations 12 and 13 to derive

$$\mathbf{V}^T\boldsymbol{\eta}\boldsymbol{\beta} = \mathbf{V}^T\bar{\mathbf{x}} - \sum_{j=0}^{m-1} \frac{\mathbf{q}_j^T\mathbf{b}}{\mathbf{q}_j^T\mathbf{u}_j}\mathbf{e}_{j+1} - \boldsymbol{\beta} + \sum_{j=0}^{m-1} \mathbf{e}_{j+1}\frac{\mathbf{q}_j^T\mathbf{U}}{\mathbf{q}_j^T\mathbf{u}_j}\boldsymbol{\beta}. \tag{15}$$

which is then substituted in the last term of equation 14. This gives us $\mathbf{K}(\bar{\mathbf{x}} - \boldsymbol{\eta}\boldsymbol{\beta}) = \mathbf{b}$. ■

In the above theorem, solutions to equations 10 and 11 exist since their RHSs are in the range of $\mathbf{K}_0$. This can be seen by showing that the RHSs of these equations are orthogonal to the $\mathbf{q}_i$, the eigenvector belonging to $null(\mathbf{K}_0)$. As for the solution to equation 13, we need more discussion. We will show that a solution to equation 13 always exists in the two cases of interest (in this paper) – the Poisson equation with Neumann and Dirichlet boundary conditions respectively – by proving that the capacitance matrix $\mathbf{C}$ is nonsingular.

For the surface reconstruction and SFS problems, we choose $\mathbf{K}_0$ to be the matrix corresponding to the discrete Laplacian operator on an embedded rectangular domain with periodic boundary conditions. For this particular choice, $rank(\mathbf{K}_0) = (N-1)$. In the case of Poisson equation with Dirichlet boundary condition (e.g., SR and SFS problems) the stiffness matrix $\mathbf{K}$ is nonsingular and the following theorem concludes that the capacitance matrix $\mathbf{C}$ will be nonsingular implying that the equation 13 always has a solution.

Corollary 1 *In theorem 1, if* $rank(\mathbf{K}) = N$ *and* $rank(\mathbf{K}_0) = N-1$*, then the matrix* $\mathbf{C}$ *is nonsingular.*

Proof: To prove that $\mathbf{C}$ is nonsingular, we show that $\mathbf{C}\boldsymbol{\beta} = \mathbf{0} \Rightarrow \boldsymbol{\beta} = \mathbf{0}$. When $m = 1$, we have the condition $rank(\mathbf{K}_0) = (N-1)$. Suppose $\mathbf{C}\boldsymbol{\beta} = \mathbf{0}$, then we have

$$\mathbf{V}^T\boldsymbol{\eta}\boldsymbol{\beta} = \mathbf{e}_1\frac{\mathbf{q}_0^T\mathbf{U}\boldsymbol{\beta}}{\mathbf{q}_0^T\mathbf{u}_0} - \boldsymbol{\beta} \tag{16}$$

from equation 12. By definition, $\mathbf{K}\boldsymbol{\eta}\boldsymbol{\beta} = \mathbf{K}_0\boldsymbol{\eta}\boldsymbol{\beta} + \mathbf{U}\mathbf{V}^T\boldsymbol{\eta}\boldsymbol{\beta}$. Substituting equations 11 and 16 into this equation, we get $\mathbf{K}\boldsymbol{\eta}\boldsymbol{\beta} = \mathbf{0}$. Since $\mathbf{K}$ is nonsingular, this implies $\boldsymbol{\eta}\boldsymbol{\beta} = \mathbf{0}$. Hence, we have $\mathbf{K}_0\boldsymbol{\eta}\boldsymbol{\beta} = \mathbf{0}$. Substituting for $\mathbf{K}_0\boldsymbol{\eta}$ from equation 11 yields,

$$\left[\begin{array}{cccc} \mathbf{0} & \mathbf{u}_1 - \frac{\mathbf{q}_0^T\mathbf{u}_1}{\mathbf{q}_0^T\mathbf{u}_0}\mathbf{u}_0 & \ldots & \mathbf{u}_{p-1} - \frac{\mathbf{q}_0^T\mathbf{u}_{p-1}}{\mathbf{q}_0^T\mathbf{u}_0}\mathbf{u}_0 \end{array} \right] \boldsymbol{\beta} = \mathbf{0}. \tag{17}$$

After rearranging, this yields

$$\sum_{i=1}^{p-1} \beta_i \mathbf{u}_i = \sum_{i=1}^{p-1} \frac{\mathbf{q}_0^T\mathbf{u}_i}{\mathbf{q}_0^T\mathbf{u}_0}\mathbf{u}_0. \tag{18}$$

The assumption that $\mathbf{u}_0, \ldots, \mathbf{u}_{p-1}$ are linearly independent combined with equation 18 makes $\beta_i = 0,\ for\ \ 1 \leq i \leq p-1$. Since, $\boldsymbol{\eta}\boldsymbol{\beta} = \sum_{i=0}^{p-1} \beta_i\boldsymbol{\eta}_i = \mathbf{0}$, we have $\beta_0\boldsymbol{\eta}_0 = \mathbf{0} \Rightarrow \beta_0 = 0$ because, $\boldsymbol{\eta}_0$ was assumed to be a nonzero vector in theorem 1. Thus, $\beta = \mathbf{0}$, and this leads to the conclusion that $\mathbf{C}$ is nonsingular. ■

Now let's consider the problem of Poisson equation with Neumann boundary condition. The matrix $\mathbf{K}_0$ is the same as that in the Dirichlet boundary condition case, but the stiffness matrix $\mathbf{K}$ is singular here. To be more precise, $rank(\mathbf{K}) = rank(\mathbf{K}_0) = N - 1$ and $null(\mathbf{K}) = null(\mathbf{K}_0) = span(\mathbf{q}_0)$. In addition, after discretizing the boundary condition, we have $\forall i, \mathbf{q}_0^T\mathbf{u}_i = \mathbf{q}_0^T\mathbf{v}_i = 0$. This leads to the following corollary.

Corollary 2 *In theorem 1, if* $rank(\mathbf{K}) = rank(\mathbf{K}_0) = N - 1$, $null(\mathbf{K}_0) = null(\mathbf{K})$ *and* $\mathbf{q}_0^T\mathbf{u}_i = \mathbf{q}_0^T\mathbf{v}_i = 0, \forall i$, *then equation 10 can be simplified as*

$$\mathbf{K}_0\bar{\mathbf{x}} = \mathbf{b}, \tag{19}$$

and equation 11 can be simplified as

$$\mathbf{K}_0\boldsymbol{\eta}_i = \mathbf{u}_i, \tag{20}$$

$0 \leq i \leq p-1$. *The capacitance matrix* $\mathbf{C} = \mathbf{I} + \mathbf{V}^T\boldsymbol{\eta}$ *is nonsingular and equation 13 becomes*

$$\boldsymbol{C}\boldsymbol{\beta} = \mathbf{V}^T\bar{\mathbf{x}}. \tag{21}$$

Therefore, $\bar{\mathbf{x}} - \boldsymbol{\eta}\boldsymbol{\beta}$ *is a solution to the linear system* $\mathbf{K}\mathbf{x} = \mathbf{b}$.

Proof: Since $\mathbf{b} \in range(\mathbf{K})$, $\mathbf{b}$ is othogonal to $null(\mathbf{K}) = null(\mathbf{K}_0)$, i.e. $\mathbf{q}_0^T\mathbf{b} = 0$. So equation 10 is simplified to 19. Similarly, $\mathbf{q}_0^T\mathbf{u}_i = 0$ for all i, therefore equation 11 is simplified to equation 20.

To prove that $\mathbf{C}$ is nonsingular, we show that $\mathbf{C}\boldsymbol{\beta} = \mathbf{0} \Rightarrow \boldsymbol{\beta} = \mathbf{0}$. Suppose $\mathbf{C}\boldsymbol{\beta} = \mathbf{0}$, it is obvious that $\mathbf{V}^T\boldsymbol{\eta}\boldsymbol{\beta} = -\boldsymbol{\beta}$. Thus we have

$$\begin{aligned}
\mathbf{K}\boldsymbol{\eta}\boldsymbol{\beta} &= \mathbf{K}_0\boldsymbol{\eta}\boldsymbol{\beta} + \mathbf{U}\mathbf{V}^T\boldsymbol{\eta}\boldsymbol{\beta} \\
&= \mathbf{K}_0\boldsymbol{\eta}\boldsymbol{\beta} + \mathbf{U}(\mathbf{V}^T\boldsymbol{\eta}\boldsymbol{\beta} \\
&= \mathbf{U}\boldsymbol{\beta} + \mathbf{U}(-\boldsymbol{\beta}) \\
&= \mathbf{0}
\end{aligned}$$

This means $\boldsymbol{\eta}\boldsymbol{\beta} \in null(\mathbf{K}) = null(\mathbf{K}_0)$. Consequently, $\mathbf{K}_0\boldsymbol{\eta}\boldsymbol{\beta} = \mathbf{0}$. Substituting for $\mathbf{K}_0\boldsymbol{\eta}$ from equation 20 yields,

$$\left[\begin{array}{cccc} \mathbf{u}_0 & \mathbf{u}_1 & \ldots & \mathbf{u}_{p-1} \end{array} \right] \boldsymbol{\beta} = \mathbf{0}. \tag{22}$$

Since $\mathbf{u}_0, \ldots, \mathbf{u}_{p-1}$ are assumed to be linearly independent, we have $\boldsymbol{\beta} = \mathbf{0}$, and this proves that $\mathbf{C}$ is nonsingular. ■

In summary, we propose theorem 1 as a general theorem for the capacitance matrix method. It can be used for Elliptic PDEs of **any order** as long as equation 13 has a solution. In this paper, we are particularly interested in solving the low-level vision problems corresponding to the 2-D Poisson equation with Dirichlet and Neumann boundary conditions. Consequently, corollaries 1 and 2 are presented to show that the capacitance matrix equation has a unique solution for the Dirichlet and Neumann cases.

3.2 The choice of $\mathbf{K}_0$

The capacitance matrix method can be used to solve several linear systems of the form $\mathbf{K}_0\mathbf{x} = \mathbf{b}$ as shown in equations 10 and 11 respectively. Especially, in equation 11, we have to solve it p times, where p is the rank of $\mathbf{K} - \mathbf{K}_0$. Therefore, it is very crucial to choose a well structured $\mathbf{K}_0$ so as to efficiently compute the solution in the capacitance matrix method.

In general, there are three criteria that dictate the choice $\mathbf{K}_0$. The first is the difference between $\mathbf{K}_0$ and $\mathbf{K}$ must be small in the sense that $rank(\mathbf{K} - \mathbf{K}_0)$ is small. Secondly, $\mathbf{K}_0$ must possess some nice structure so as to have a fast numerical solution to the associated linear system. Lastly, $\mathbf{K}_0$

should be translation invariant; this property has the advantage of saving the computational cost in solving equation 11 for each $\mathbf{u}_i, 0 \leq i \leq p-1$, since there are only one or two nonzero components in each $\mathbf{u}_i$. Therefore, we need to compute the solution $\mathbf{h}_0$ to a linear system consisting of a matrix $\mathbf{K}_0$ and an RHS vector which is the projection of the unit vector onto the range of $\mathbf{K}_0$. Because, $\mathbf{h}_0$ is translation invariant, we can obtain the solution of $\boldsymbol{\eta}_i$ in equation 11 without having to solve any of the linear systems associated with each $\boldsymbol{\eta}_i$.

For discretized Poisson equations on irregular domains, we use the standard technique of imbeding irregular domains in rectangular regions [6]. Thus, the boundary conditions can be easily incorporated into $\mathbf{U}\mathbf{V}^T$. Following the three categories given above, we can choose $\mathbf{K}_0$ as the discrete Laplacian operator on the doubly periodic imbedded rectangular domain. The exact form of $\mathbf{K}_0$ is given by.

$$\mathbf{K}_0 = \begin{bmatrix} \mathbf{D} & -\mathbf{I} & & & -\mathbf{I} \\ -\mathbf{I} & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -\mathbf{I} \\ -\mathbf{I} & & & -\mathbf{I} & \mathbf{D} \end{bmatrix} \tag{23}$$

Where,

$$\mathbf{D} = \begin{bmatrix} 4 & -1 & & & -1 \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ -1 & & & -1 & 4 \end{bmatrix}. \tag{24}$$

The matrix $\mathbf{K}_0$ can be regarded as the result obtained from the 5-point approximation to Laplacian operator on a doubly periodic imbedded rectangular domain. From a computational molecule [23] point of view, only the membrane molecule is used at every node of the doubly periodic rectangular domain. Therefore, the data constraint molecule, derivative (Neumann boundary) constraint molecule and all molecular inhibitions due to depth discontinuities and the imbeding boundary are included in the matrix $\mathbf{U}\mathbf{V}^T$. Since $\mathbf{U}\mathbf{V}^T = \sum_{i=0}^{p-1} \mathbf{u}_i \mathbf{v}_i^T$, each $\mathbf{u}_i \mathbf{v}_i^T$ can be considered as a data constraint molecule, a derivative constraint molecule or a molecule inhibition at some location. Thus, we can see that the structures of $\mathbf{u}_i$ and $\mathbf{v}_i$ are very simple. Each vector contains either one or two nonzero components. One nonzero component is for the case of the data constraint

molecule while, two nonzero components are for the other two cases.

We can see that this particular choice of $\mathbf{K}_0$ makes it circulant Toeplitz and thus translation invariant. In addition, it can be decomposed as the sum of two tensor products of matrices $\mathbf{A}$ and $\mathbf{I}$, i.e.,

$$\mathbf{K}_0 = \mathbf{A} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{A}, \tag{25}$$

with

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & & & -1 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ -1 & & & -1 & 2 \end{bmatrix} \in \Re^{n \times n},$$

and $\otimes$ is the tensor (Kronecker) product. By using this special structure of the matrix $\mathbf{K}_0$, we can rewrite any linear system of the form $\mathbf{K}_0\mathbf{z} = \mathbf{f}$ as the following Lyapunov matrix equation

$$\mathbf{AZ} + \mathbf{ZA} = \mathbf{F}, \tag{26}$$

where $\mathbf{Z}$ and $\mathbf{F}$ are $n \times n$ matrices corresponding to their concatenated $n^2 \times 1$ vectors $\mathbf{z}$ and $\mathbf{f}$ respectively. It should be noted that $n^2 = N$, and that the matrix $\mathbf{A}$ in equation 26 is circulant Toeplitz and symmetric positive-semidefinite. We can use the ADI method to solve this Lyapunov matrix equation in $O(N)$ operations. We will prove this $O(N)$ computational complexity of the ADI method in the next section.

3.3 Structure of the proposed Algorithm

In section 3.1, we proved that the capacitance matrix technique can be used to solve the discretized Poisson equation with Dirichlet and Neumann boundary conditions. We now describe the structure of the proposed algorithm via a flowchart given in figure 1.

The flow of the algorithm follows from theorem 1. For linear systems with matrix $\mathbf{K}_0$, we use the ADI method to solve the corresponding Lyapunov matrix equations. To form the capacitance matrix $\mathbf{C}$, we have to know $\boldsymbol{\eta}_i, 0 \leq i \leq p-1$, in equation 11. In fact, we don't need to solve equation 11 for each $\boldsymbol{\eta}_i$. By using the translation invariance property of $\mathbf{K}_0$, we just solve the linear system (using the ADI method)

$$\mathbf{K}_0\mathbf{h}_0 = \mathbf{e}_1 - \frac{\mathbf{q}_0^T\mathbf{e}_1}{\mathbf{q}_0^T\mathbf{q}_0}\mathbf{q}_0, \tag{27}$$

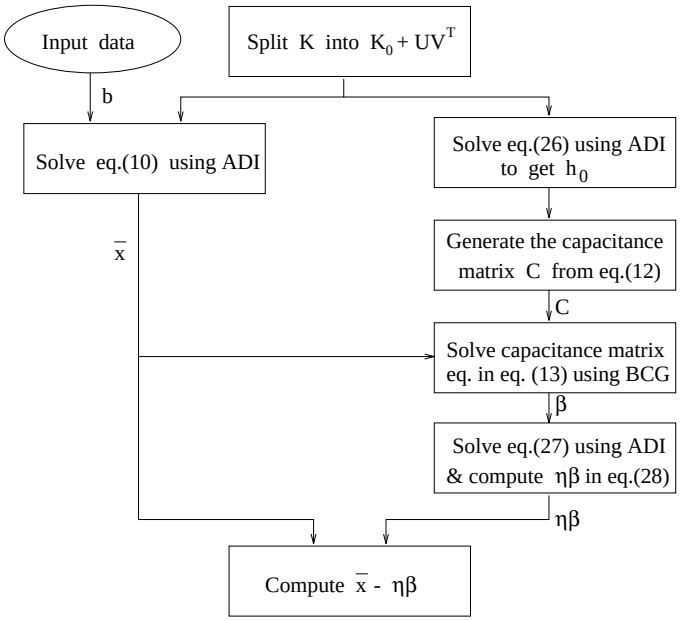

Figure 1: The flow chart of the proposed algorithm

where the unit vector $\mathbf{e}_1$ has a single nonzero element at location 1. Thus, the solution $\mathbf{h}_0$ can be regarded as the impulse response to the pseudo-inverse operator $\mathbf{K}_0^+$. Since $\mathbf{K}_0$ is singular with a simple zero eigenvalue, the general form of the solutions can be written as $\mathbf{h}_0 + a\mathbf{q}_0, \forall a \in \Re$. The solution $\boldsymbol{\eta}_i$ can be easily obtained from the impulse response $\mathbf{h}_0$ because $\mathbf{K}_0^+$ is a linear shift invariant operator. In fact, we don't need to explicitly compute and store every $\boldsymbol{\eta}_i$ since, they are needed only for the computation of the $\mathbf{C}$ matrix using equation 12. This is achieved by obtaining the (i,j)-th component of $\mathbf{V}^T\boldsymbol{\eta}$ i.e., $\mathbf{v}_i^T\boldsymbol{\eta}_j$ from the values of $\mathbf{h}_0$ corresponding to the relative locations of the nonzero components of $\mathbf{v}_i$ and $\mathbf{u}_j$.

In addition, we don't explicitly compute $\boldsymbol{\eta\beta}$ for the final solution since this matrix-vector multiplication takes $O(pn^2) = O(n^3)$ operations (as $p = O(n)$). Instead, we solve the following linear system

$$\mathbf{K}_0\mathbf{y} = \mathbf{U}\boldsymbol{\beta} - \frac{\mathbf{q}_0^T\mathbf{U}\boldsymbol{\beta}}{\mathbf{q}_0^T\mathbf{u}_0}\mathbf{u}_0 \tag{28}$$

to get the solution $\mathbf{y}$ which has the same projection on the range of $\mathbf{K}_0$ as $\boldsymbol{\eta\beta}$. As for the projection of $\boldsymbol{\eta\beta}$ on the null space of $\mathbf{K}_0$, it can be easily computed from the inner product of $\boldsymbol{\beta}$ and $\mathbf{a} = (a_0, \ldots, a_{p-1})$, where $a_i = \mathbf{q}_0^T\boldsymbol{\eta}_i$ is the projection of $\boldsymbol{\eta}_i$ on the null space of $\mathbf{K}_0$. The coefficients a_i can be chosen arbitrarily as long as $a_0 \neq 0$. To simplify the computations in our implementation,

we assign $a_i = 1, \forall i$. Thus,

$$\boldsymbol{\eta\beta} = \mathbf{y} + (\mathbf{a}^T\boldsymbol{\beta} - \frac{\mathbf{q}_0^T\mathbf{y}}{\mathbf{q}_0^T\mathbf{q}_0})\mathbf{q}_0. \tag{29}$$

The capacitance matrix $\mathbf{C}$ is a dense matrix of size $p \times p$. To solve for $\boldsymbol{\beta}$ in equation 13, we employ the BCG (biconjugate gradient) method in a wavelet basis to approximate the matrix-vector multiplication in each iteration. The details of this modified BCG method and the ADI method are discussed in the next section.

4 Numerical Solution

In this section, we discuss the convergence of ADI method applied to the Lyapunov matrix equation and show that the computational cost for solving this equation is $O(N)$ by proving that the ADI method can converge in a constant number of iterations independent of N. In addition, we present a modified biconjugate gradient method that is used in solving the capacitance linear system.

4.1 ADI Method for the Lyapunov Equation

The ADI method for solving a Lyapunov matrix equation is described in [16]. For any Lyapunov matrix equation of the form $\mathbf{AX} + \mathbf{XA} = \mathbf{B}$, the ADI method involves the following steps in each iteration $j = 1, 2, \ldots, J$,

$$(\mathbf{A} + p_j\mathbf{I})\mathbf{X}_{j-\frac{1}{2}} = \mathbf{B} - \mathbf{X}_{j-1}(\mathbf{A} - p_j\mathbf{I}) \tag{30}$$

$$\mathbf{X}_j(\mathbf{A} + p_j\mathbf{I}) = \mathbf{B} - (\mathbf{A} - p_j\mathbf{I})\mathbf{X}_{j-\frac{1}{2}} \tag{31}$$

For our problem, note that the matrix $\mathbf{A}$ is circulant Toeplitz and symmetric positive semidefinite as discussed in section 3.2. The special structure of matrix A helps in advancing the ADI iterations very efficiently in each step of the technique. On the left-hand side (LHS) of the equations 30 and 31 respectively, the matrix $\mathbf{A} + p_j\mathbf{I}$ is very close to being tridiagonal except for the nonzero entries at the corners namely, the entries (1,n) and (n,1). These nonzero corner entries are caused due to the doubly periodic boundary conditions discussed in the last section. In addition, $\mathbf{A} + p_j\mathbf{I}$ is SPD for any positive parameter p_j. Therefore, we can compute the LU decomposition of this matrix

and then use the forward and back substitution to compute the updated solution. Due to the the special structure of the matrix $\mathbf{A} + p_j\mathbf{I}$, the solution update requires only $O(N)$ time per iteration.

We now examine the convergence of the ADI method for the Lyapunov matrix equation in our algorithm. Let the initial $\mathbf{X}_0 = \mathbf{0}$ (zero matrix), and $\triangle\mathbf{X}_t = \mathbf{X}_t - \mathbf{X}^*$, where $\mathbf{X}^*$ is the true solution of the Lyapunov matrix equation, $\lambda_0, \lambda_1, \ldots, \lambda_{n-1}$, and $\mathbf{q}_0, \mathbf{q}_1, \ldots, \mathbf{q}_{n-1}$ be the eigenvalues and the eigenvectors of $\mathbf{A}$ respectively. For this symmetric matrix $\mathbf{A}$, we have

$$\mathbf{A} = \mathbf{Q}\mathbf{D}\mathbf{Q}^T, \tag{32}$$

where $\mathbf{Q} = [\mathbf{q}_0 \quad \mathbf{q}_1 \quad \cdots \quad \mathbf{q}_{n-1}]$, $\mathbf{D} = diag[\lambda_0, \lambda_1, \cdots, \lambda_{n-1}]$, and $\lambda_k = 4\sin^2(\frac{\pi\lfloor\frac{k+1}{2}\rfloor}{n})$ for $k = 0, \ldots, n-1$. We express the eigenvalue λ_k in this form so as to have nondecreasing eigenvalues with respect to the indices. The range of eigenvalues is between 0 and 4, and the zero eigenvalue (when $k = 0$) is simple. The orthogonal matrix $\mathbf{Q}$ contains the discrete sine or cosine function in each column with increasing frequency corresponding to increasing eigenvalue, i.e., the index increases. For example, the eigenvector corresponding to the simple zero eigenvalue is a flat function whose frequency is 0.

By taking the tensor product of the eigenvectors, $\mathbf{q}_k \otimes \mathbf{q}_l, k, l = 0, \ldots, n-1$, we can generate an orthogonal basis for $\Re^{n\times n}$ and express the true solution $\mathbf{X}^*$ in this basis as follows:

$$\mathbf{X}^* = \sum_{k=0}^{n-1}\sum_{l=0}^{n-1} a_{kl}(\mathbf{q}_k \otimes \mathbf{q}_l). \tag{33}$$

Subtracting each of the equations 30 and 31 from the Lyapunov equation $\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{A} = \mathbf{B}$, and expressing $\mathbf{X}$ in the above tensor product basis, we have the following result for the error after t iterations,

$$\triangle\mathbf{X}_t = -\sum_{k=0}^{n-1}\sum_{l=0}^{n-1}\left[\prod_{j=1}^{t}(\frac{\lambda_k - p_j}{\lambda_k + p_j})(\frac{\lambda_l - p_j}{\lambda_l + p_j})\right] a_{kl}(\mathbf{q}_k \otimes \mathbf{q}_l). \tag{34}$$

The function inside the bracket is the error reduction factor in the basis $\mathbf{q}_k \otimes \mathbf{q}_l$. Notice that the Lyapunov matrix equation has an infinite number of solutions due to the singularity of the matrix $\mathbf{A}$. The error reduction in each step is always less than 1 for eigenvectors with positive eigenvalues and positive parameters p_j while it is always 1 for the basis $\mathbf{q}_0 \otimes \mathbf{q}_0$ (which is evident from equation 34 after substituting $k = l = 0$ and $\lambda_0 = 0$). A solution to the Lyapunov matrix equation can be obtained without reducing this factor at the component $k = l = 0$.

The classical ADI minimax parameter problem is to find the ADI parameters p_j to minimize the maximum of the error reduction function excluding the zero eigenvalue, i.e., to minimize the function

$$\mu_t = \max_{\forall(k,l)\backslash(0,0)} |\prod_{j=1}^{t} (\frac{\lambda_k - p_j}{\lambda_k + p_j})(\frac{\lambda_l - p_j}{\lambda_l + p_j})|. \tag{35}$$

Applying the result from the classical ADI minimax analysis [16], leads to an $O(\log N)$ iterations for convergence [24]. This in turn yields an overall computational complexity of $O(N \log N)$ for the ADI method since, $O(N)$ time is required per iteration in the ADI method. An $O(N \log N)$ computational complexity is unacceptable for our purposes (recall that we seek an $O(N)$ solution). Thus, in order to design an $O(N)$ algorithm, it is essential to use the additional knowledge about the problem e.g., spectral characteristics of the differential operator in the PDE. The classical ADI convergence analysis does not make use of any such knowledge. In this thesis, we reformulate the ADI convergence analysis by incorporating the spectral characteristics of the Laplacian operator. This is facilitated by making the following assumption.

Assumption 1 *Let the projection of* $\mathbf{B}$*, the RHS of the Lyapunov matrix equation, on the basis* $\mathbf{q}_k \otimes \mathbf{q}_l$ *be* b_{kl}*. Assume there exist a constant* $\alpha \in \Re^+$ *and a* $w_m > 0$ *such that ,*$\forall(k,l)\backslash(0,0), |b_{kl}| \leq \frac{w_m}{(k^2+l^2)^m}$ *for some* $m > -0.5$*, and* $\sum_k \sum_l (\frac{b_{kl}}{k^2+l^2})^2 \geq \alpha w_m^2$.

For our Lyapunov matrix equation $\mathbf{AX} + \mathbf{XA} = \mathbf{B}$, the relationship between the solution $\mathbf{X}$ and the data $\mathbf{B}$ in the eigen basis of the matrix $\mathbf{K}_0$ is $a_{kl} = \frac{b_{kl}}{\lambda_k+\lambda_l}$. Therefore, the assumption $|b_{kl}| \leq \frac{w_m}{(k^2+l^2)^m} \Rightarrow |a_{kl}| \leq \frac{w_m}{(\lambda_k+\lambda_l)(k^2+l^2)^m}$ for some $m > -0.5$. Using the relation $\frac{2}{\pi}x \leq \sin x \leq x$ for $x \in [0, \pi/2]$, we have $c_1 k^2 \leq \lambda_k \leq c_2 k^2$ with $c_1 = \frac{4}{n^2}$ and $c_2 = \frac{4\pi^2}{n^2}$. Thus, requiring that the solution satisfy the constraint $|a_{kl}| \leq \frac{w_m n^2}{(k^2+l^2)^{m+1}}$ for some $m > -0.5$.

This assumption can be given the following spectral interpretation. The smaller values k or l correspond to lower frequencies along vertical or horizontal direction, and vice versa. Consequently, the assumption means that the overall frequency content of the input data or the right hand side can be bounded by an increasing function of $\|\omega\|_2 = (k^2 + l^2)^{0.5}$ of degree less than 1. This can be easily satisfied when the overall frequency distribution of the input data is decreasing or uniform. This is true in reality since, the low frequency content in signals is normally much higher than the high frequency content.

We first define two matrices $\mathbf{X}'$ and $\triangle\mathbf{X}'_t$ for use in the convergence analysis of the ADI method. Let $\mathbf{X}'$ and $\triangle\mathbf{X}'_t$ be the projections of the true solution $\mathbf{X}^*$ and the error in computed solution (in iteration t) $\triangle\mathbf{X}_t$ on the range of the matrix $\mathbf{K}_0$ ($= \mathbf{A}\otimes\mathbf{I} + \mathbf{I}\otimes\mathbf{A}$), respectively. Note that, $\mathbf{X}'$ and $\mathbf{X}^*$ are almost the same except that $\mathbf{X}'$ doesn't have projection at the basis $\mathbf{q}_0\otimes\mathbf{q}_0$. In the ADI literature [16], the convergence criterion is normally defined as follows: The ADI method for the Lyapunov matrix equation is said to have converged when the relative Frobenius norm $\frac{\|\triangle\mathbf{X}'_t\|_F}{\|\mathbf{X}'\|_F}$ is reduced to within a specified error tolerance ϵ lying between 0 and 1, i.e., $\|\triangle\mathbf{X}'_t\|_F \leq \epsilon\|\mathbf{X}'\|_F$. This means we don't need to reduce the error component in the basis corresponding to the zero eigenvalue or in the null space of $\mathbf{K}_0$. The problem is to find the number of iterations required in the ADI method and the associated ADI parameter set to achieve $\|\triangle\mathbf{X}'_t\|_F \leq \epsilon\|\mathbf{X}'\|_F$.

We make use of this convergence criterion and assumption 1 to reformulate the convergence theory of the ADI method as follows. The Frobenius norm $\|\triangle\mathbf{X}'_t\|_F$ can be written as,

$$\|\triangle\mathbf{X}'_t\|_F^2 = \sum_k\sum_l a_{kl}^2 \prod_{j=1}^{t}(\frac{\lambda_k - p_j}{\lambda_k + p_j})^2(\frac{\lambda_l - p_j}{\lambda_l + p_j})^2 \tag{36}$$

where the double summation takes every k and l between 0 and $n-1$ except $k = l = 0$ (simultaneously). As a consequence of assumption 1 the ADI method for our Lyapunov matrix equation can converge in a constant number of iterations. We state and prove this convergence result in the following theorem.

Theorem 2 *Given assumption 1 and an error tolerance ϵ between 0 and 1, there exists a constant integer t such that $\forall n$, $\|\triangle\mathbf{X}'_t\|_F \leq \epsilon\|\mathbf{X}'\|_F$.*

Proof: From the discussion on assumption 1, we have the following inequality for $\|\mathbf{X}'\|_F$.

$$\|\mathbf{X}'\|_F^2 = \sum_k\sum_l(\frac{b_{kl}}{\lambda_k + \lambda_l})^2 \geq \frac{n^4}{16}\sum_k\sum_l(\frac{b_{kl}}{k^2 + l^2})^2 \geq \frac{n^4}{16}\alpha w_m^2 \tag{37}$$

Applying assumption 1 to equation 36, we have the following inequality

$$\begin{aligned}\|\triangle\mathbf{X}'_t\|_F^2 &\leq \sum_k\sum_l \frac{w_m^2}{(k^2+l^2)^{2m}(\lambda_k+\lambda_l)^2}\prod_{j=1}^{t}(\frac{\lambda_k - p_j}{\lambda_k + p_j})^2(\frac{\lambda_l - p_j}{\lambda_l + p_j})^2 \\ &= \frac{w_m^2}{n^{4m-2}}\sum_k\sum_l \frac{1}{((\frac{k}{n})^2+(\frac{l}{n})^2)^{2m}(\lambda_k+\lambda_l)^{2-q}}(\frac{1}{n^2})R(p_1,p_2,\ldots,p_t),\end{aligned}$$

where q is a positive number between 0 and $2m+1$ and

$$R(p_1, p_2, \ldots, p_t) = \frac{1}{(\lambda_k + \lambda_l)^q} \prod_{j=1}^{t} (\frac{\lambda_k - p_j}{\lambda_k + p_j})^2 (\frac{\lambda_l - p_j}{\lambda_l + p_j})^2. \tag{38}$$

Using the fact that $\lambda_k = 4\sin^2(\frac{\pi[\frac{k+1}{2}]}{n}) \geq 4(\frac{k}{n})^2$ for $k \geq 0$ and substituting into the above inequality, we have

$$\|\triangle \mathbf{X'}_t\|_F^2 \leq \frac{w_m^2}{4^{2-q} n^{4m-2}} \sum_k \sum_l \left\{ \frac{1}{((\frac{k}{n})^2 + (\frac{l}{n})^2)^{2m+2-q}} (\frac{1}{n^2}) R(p_1, p_2, \ldots, p_t) \right\}, \tag{39}$$

Then we have,

$$\|\triangle \mathbf{X'}_t\|_F^2 \leq \frac{w_m^2}{4^{2-q} n^{4m-2}} \sum_k \sum_l \left\{ \frac{1}{((\frac{k}{n})^2 + (\frac{l}{n})^2)^{2m+2-q}} \frac{1}{n^2} \right\} \max_{\forall (k,l) \backslash (0,0)} R(p_1, p_2, \ldots, p_t). \tag{40}$$

Note that the double summation on the RHS of equation 40 takes every integer k and l between 0 and $n-1$ except $k = l = 0$ simultaneously. Let the set of all the pair (k, l) in this double summation be denoted by A. By using the split in the set A, we have

$$\sum_k \sum_l f(k,l) \frac{1}{n^2} = \frac{1}{n^2} \{ f(0,1) + f(1,0) + f(1,1) + \sum_{k=2}^{n-1} f(k,0) + \sum_{l=2}^{n-1} f(0,l) + \sum_{(k,l) \in B} f(k,l) \}, \tag{41}$$

where $f(k,l) = \frac{1}{((\frac{k}{n})^2 + (\frac{l}{n})^2)^{2m+2-q}}$ and $B = (k,l) \backslash (1,1) \mid k, l = 1, \ldots, n-1$. By using the fact that $f(k,l)$ is a strictly decreasing function of k and l, we can bound the terms with summation in equation 41 by the corresponding terms in an integration form as follows.

$$\begin{aligned} \sum_k \sum_l f(k,l) \frac{1}{n^2} &< \{ f(0,1) + f(1,0) + f(1,1) \} \frac{1}{n^2} + \frac{1}{n} \int_{\frac{1}{n}}^{\frac{n-1}{n}} \frac{1}{x^{4m+4-2q}} \, dx \\ &\quad + \frac{1}{n} \int_{\frac{1}{n}}^{\frac{n-1}{n}} \frac{1}{y^{4m+4-2q}} \, dy + \int \int_S \frac{1}{(x^2+y^2)^{2m+2-q}} \, dxdy, \end{aligned} \tag{42}$$

where the integration domain $S = \{(x,y) | (x,y) \in [0,1] \times [0,1] \backslash [0, \frac{1}{n}) \times [0, \frac{1}{n})\}$. By transforming the Cartesian coordinate to the polar coordinate, the term with double integral in equation 42 can be bounded by the following integral,

$$\begin{aligned} \int_0^{\frac{\pi}{2}} \int_{\frac{1}{n}}^{\sqrt{2}} \frac{1}{r^{4m+4-2q}} r \, dr d\theta &= \frac{\pi}{4(2m+1-q)} (n^{4m+2-2q} - 2^{2m+1-q}) \\ &< \frac{\pi}{4(2m+1-q)} n^{4m+2-2q}. \end{aligned} \tag{43}$$

Substituting equation 43 into equation 42 and calculating the integrals in equation 42, we obtain the following inequality

$$\sum_k \sum_l \frac{1}{((\frac{k}{n})^2 + (\frac{l}{n})^2)^{2m+2-q}} \frac{1}{n^2} < \beta n^{4m+2-2q}, \tag{44}$$

where $\beta = 2 + \frac{1}{2^{2m+2-q}} + \frac{2}{4m+3-2q} + \frac{\pi}{4(2m+1-q)}$ is a constant.

Using equations 37, 40 and 44, the convergence criterion $\|\triangle \mathbf{X}_t\|_F \leq \epsilon \|\mathbf{X}'\|_F$ can be met with by the following derived condition

$$\max_{\forall (k,l) \setminus (0,0)} R(p_1, p_2, \ldots, p_t) \leq \frac{\alpha \epsilon^2 4^{-q}}{\beta} n^{2q} = M n^{2q}, \tag{45}$$

where M is the constant defined as $M := \frac{\alpha \epsilon^2 4^{-q}}{\beta}$ Furthermore, we can bound the function $\max_{k,l \setminus (0,0)} R(p_1, p_2, \ldots$ by a function of λ_k as follows:

$$\begin{aligned} \max_{\forall (k,l) \setminus (0,0)} R(p_1, \ldots, p_t) &= \max_{\forall (k,l) \setminus (0,0)} \left\{ \frac{1}{(\lambda_k + \lambda_l)^q} \prod_{j=1}^{t} (\frac{\lambda_k - p_j}{\lambda_k + p_j})^2)(\prod_{j=1}^{t} (\frac{\lambda_l - p_j}{\lambda_l + p_j})^2 \right\} \\ &\leq \max_{1 \leq k \leq n-1} \frac{1}{\lambda_k^q} \prod_{j=1}^{t} (\frac{\lambda_k - p_j}{\lambda_k + p_j})^2 \end{aligned}$$

The maximum taken from the values of function R at the nonzero eigenvalues,can be bounded by using the maximum of the same function in an interval containing all the nonzero eigenvalues. Then the convergence can be reached by using the following modified requirement

$$\max_{\lambda_1 = 4 \sin^2(\frac{\pi}{n}) \leq x \leq 4} \frac{1}{x^q} \prod_{j=1}^{t} (\frac{x - p_j}{x + p_j})^2 \leq M n^{2q}. \tag{46}$$

For any positive p_j, the function $\frac{1}{x^q} \prod_{j=1}^{t} (\frac{x-p_j}{x+p_j})^2$ is always less than the right-hand side when $x \geq \frac{1}{M^{\frac{1}{q}} n^2}$. Therefore, in equation 46, the interval $(4 \sin^2(\frac{\pi}{n}), 4)$ can be replaced by $S = (4 \sin^2(\frac{\pi}{n}), min(4, \frac{1}{M^{\frac{1}{q}} n^2}))$. The convergence requirement in equation 46 can be made less stringent by using the inequality,

$$\max_{x \in S} \frac{1}{x^q} \prod_{j=1}^{t} (\frac{x - p_j}{x + p_j})^2 \leq (\max_{x \in S} \prod_{j=1}^{t} (\frac{x - p_j}{x + p_j})^2)(\max_{x \in S} \frac{1}{x^q}), \tag{47}$$

leading to the following modified requirement which when satisfied automatically makes the inequality 46 true.

$$\max_{x \in S} \prod_{j=1}^{t} (\frac{x - p_j}{x + p_j})^2 \leq (M n^{2q})(\frac{1}{\max_{x \in S} \frac{1}{x^q}}) = (M n^{2q})(4 \sin^2(\frac{\pi}{n}))^q = M' \tag{48}$$

This bound M' will approach a constant when $n \to \infty$. Thus, the problem becomes one of finding the number of iterations t and the parameters p_j such that the requirement in 48 is satisfied. This requirement is of the same form as the classical ADI minimax problem which was solved by Jordan as quoted in [27]. We can use the result in [27] to determine the number of iterations required for convergence. The number of iterations needed to meet this requirement is

$$J = \lceil \frac{ln\frac{4}{M'} ln\frac{4}{k'}}{\pi^2} \rceil, \tag{49}$$

$$k' = \frac{1}{\nu' + \sqrt{\nu'^2 - 1}}, \tag{50}$$

$$\nu' = \frac{1}{2}(\nu + \frac{1}{\nu}), \tag{51}$$

where, $\lceil z \rceil$ denotes the smallest integer larger than z and ν is the expansion factor of an interval (a, b) defined as $\frac{b}{a}$. The expansion factor ν of the interval of interest (S) is fixed when n approaches infinity, i.e.

$$\lim_{n\to\infty} \nu = \frac{1}{4\pi^2 M^{1/q}}. \tag{52}$$

In the classical ADI minimax formulation, ν is the spectral radius of the eigenvalues being considered in $\mathbf{A}$, which is shown be of the order of n^2 [24]. Since we use a different formulation for convergence and impose the spectral assumption on the data, the resulting ν becomes $\frac{min(4, \frac{1}{Mn^2})}{4\sin^2(\frac{\pi}{n})}$. When $n \to \infty$, the ratio ν approaches a constant as given in equation 52 and M' also approaches a constant as mentioned above. Therefore, we conclude that $\lim_{n\to\infty} J$ is a constant, i.e. there is a constant t such that $\|\triangle \mathbf{X}'_t\|_F \leq \epsilon \|\mathbf{X}'\|_F$ is satisfied for all n. ■

In [27], an optimal set of ADI parameters for the general ADI iterations was given. We may use these parameters to solve the minimax problem in equation 48. Although this set of ADI parameters is not the optimal set to meet our convergence requirement, it provides us with a parameter set which can achieve the convergence criterion in a constant number of iterations. Therefore, we use the result in [27] to select a suboptimal ADI parameter set. Solving for the optimal ADI parameters in our problem requires complicated analysis using transformation theory [27] and is currently under investigation.

4.2 Modified BCG Method for the Capacitance Matrix Equation

In this section we present an algorithm to solve the capacitance matrix equation defined earlier. The conjugate-gradient(CG) method is a very powerful iterative scheme to solve the symmetric positive definite linear system. A natural generalization of the CG-type method to solve a general nonsymmetric linear system is called the biconjugate gradient(BCG) method [14]. The linear system with the capacitance matrix is a nonsymmetric and indefinite system. The size of the capacitance matrix is determined by the rank of the matrix $\mathbf{U}\mathbf{V}^T$, which contains the difference between $\mathbf{K}$ and $\mathbf{K}_0$. With the $\mathbf{K}_0$ proposed in section 3.2, the size of the associated capacitance matrix is usually $O(n)$. The BCG method requires the computation of a matrix-vector product in each iteration. Since the capacitance matrix $\mathbf{C}$ is dense, this matrix-vector product takes $O(N)$ operations. In this section, we present a modified biconjugate gradient method to solve this linear system by applying the wavelet transform technique to approximate the multiplication and reduce the computational cost to $O(n)$. Since the BCG converges in $O(n)$ iterations, the computational complexity for the modified BCG method is $O(n^2) = O(N)$.

The BCG algorithm [14] for the nonsymmetric linear system $\boldsymbol{C\beta} = \boldsymbol{f}$ is as follows.

0. Choose $\boldsymbol{\beta}_0$, and set $\mathbf{q}_0 = \mathbf{r}_0 = \mathbf{f} - \mathbf{C}\boldsymbol{\beta}_0$; Choose $\tilde{\mathbf{r}}_0 \neq \mathbf{0}$, and set $\tilde{\mathbf{q}}_0 = \tilde{\mathbf{r}}_0$;

1. Compute $\alpha_k^N = \tilde{\mathbf{r}}_{k-1}^T \mathbf{r}_{k-1}$, $\alpha_k^D = \tilde{\mathbf{q}}_{k-1}^T \mathbf{C}\mathbf{q}_{k-1}$, and $\alpha_k = \alpha_k^N / \alpha_k^D$;

2. Update $\boldsymbol{\beta}_k = \boldsymbol{\beta}_{k-1} + \alpha_k \mathbf{q}_{k-1}$;

3. Set $\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k \mathbf{C}\mathbf{q}_{k-1}$, and $\tilde{\mathbf{r}}_k = \tilde{\mathbf{r}}_{k-1} - \alpha_k \mathbf{C}^T \tilde{\mathbf{q}}_{k-1}$;

4. Compute $\rho_k^N = \tilde{\mathbf{r}}_k^T \mathbf{r}_k$, and $\rho_k = \rho_k^N / \alpha_k^N$;

5. Set $\mathbf{q}_k = \mathbf{r}_k + \rho_k \mathbf{q}_{k-1}$, and $\tilde{\mathbf{q}}_k = \tilde{\mathbf{r}}_k + \rho_k \tilde{\mathbf{q}}_{k-1}$;

6. If $\mathbf{r}_k \approx \mathbf{0}$ or $\tilde{\mathbf{r}}_k \approx \mathbf{0}$, stop; else go to 1.

Like the CG, the operations and storage requirement per step in the BCG are constant. The convergence behavior is similar to the CG method except that the BCG algorithm is susceptible to

breakdowns and numerical instabilities. To be more specific, division by 0 (or a number very close to 0) may occur in computing α_k and ρ_k. These two different breakdowns may be avoided by using the look-ahead Lanczos algorithm or the quasi-minimal residual(QMR) algorithm [8].

In the above algorithm, two matrix-vector multiplications $\mathbf{C}\mathbf{q}_{k-1}$ and $\mathbf{C}\mathbf{q}_{k-1}$ are required in each iteration. Since the matrix $\mathbf{C}$ is dense, the direct multiplication will take $O(N)$ operations, which is not acceptable to our pursuit of an $O(N)$ algorithm for the Poisson equation. Thus, we propose to carry out the multiplication in a wavelet basis to reduce the computational cost [1, 2]. With appropriate wavelet basis [2], the decomposed matrix can be approximated to high accuracy by a sparse matrix. After the approximation, the multiplication of this sparse matrix and the decomposed vector can be reduced to $O(n)$ or $O(n\log(n))$ operations [2]. Consequently, the modified BCG method is computationally efficient.

The wavelet transform of a matrix $\mathbf{D}$ and vector $\mathbf{q}$ can be written as follows:

$$\bar{\mathbf{D}} = \boldsymbol{\Phi}\mathbf{D}\boldsymbol{\Phi}^{\mathbf{T}}, \quad \bar{\mathbf{q}} = \boldsymbol{\Phi}\mathbf{q}, \tag{53}$$

where $\boldsymbol{\Phi}$ is the wavelet transform matrix with each row corresponding to a wavelet basis vector. If we use the orthogonal wavelet transform, then $\mathbf{D}\mathbf{q} = \boldsymbol{\Phi}^T(\bar{\mathbf{D}}\bar{\mathbf{q}})$. Therefore, we compute matrix-vector multiplication in the wavelet domain and then transform back to the original domain by using the wavelet reconstruction (inverse wavelet transform) to obtain the product $\mathbf{D}\mathbf{q}$.

It is very crucial to know the structure of the capacitance matrix to achieve an accurate and efficient approximation of the matrix. In equation 12, the capacitance matrix is composed of three matrices, i.e., the identity matrix $\mathbf{I}$, $-\sum_{j=0}^{m-1}\mathbf{e}_{j+1}\frac{\mathbf{q}_j^T\mathbf{U}}{\mathbf{q}_j^T\mathbf{u}_j}$ and $\boldsymbol{V}^T\boldsymbol{\eta}$. The first two matrices are extremely sparse, but the third is a dense matrix. When computing the matrix-vector product of the capacitance matrix and a vector, we directly compute the separate products for the first two sparse matrices while the aforementioned wavelet approximation scheme is used only for the matrix $\boldsymbol{V}^T\boldsymbol{\eta}$.

5 Experimental Results

In this section, we present the results of applying our algorithm to the surface reconstruction and shape from shading problems. Our algorithm can also be applied to other low-level vision problems such as the lightness, and optic flow problems, however, due to lack of space we chose to limit our discussion to the aforementioned two problems.

Surface Reconstruction: In this problem we applied our algorithm to solve the linear system $\mathbf{Kx} = \mathbf{b}$ which was derived in section 2.1.1. We synthesized a sparse range data set on a 64×64 grid and introduced a depth discontinuity along the line between coordinates $(1, 32)$ and $(30, 32)$. The input data set is very sparse and contains 15 data points randomly distributed in the plane as shown in figure 2(a). We apply our algorithm to this data to obtain the interpolated surface. We can apply equation 49 to compute the number of iterations required to guarantee the error tolerance ϵ to be within 10^{-5}. In this example we observe that 10 ADI iterations are needed to attain the tolerance. We depict the surfaces after 1 and 10 ADI iterations in figures 2(b) and 2(c) respectively.

In figure 2(b), we can see that just one ADI iteration can produce the global shape of the exact surface. This is because we advance the ADI iterations with the parameters p_j's starting with a small value and proceeding toward large values. The smaller parameters are used to recover the low frequency detail of the surface and the large values refine the shape with high frequency components. Consequently, it is not surprising that the global shape was recovered in one iteration.

This phenomena of global to local shape recovery is quite different from the other iterative numerical methods. Most of the other numerical methods do not exhibit this type of spectral characteristic during the iterations. In addition to the fast convergence rate, another advantage of the ADI method is that we can make use of the spectral properties of the differential operator in the PDE (in our case, the Laplacian) to speed up the convergence rate.

The SFS Problem: In this problem, we tested our algorithm on a synthetically generated image of a Lambertian sphere illuminated by a distant point source and viewed from the same location as the point source. Figure 3(a) depicts the 64×64 original shaded image with an 8 bit resolution in gray per pixel. The sphere is assumed to have Lambertian reflectance properties with a constant

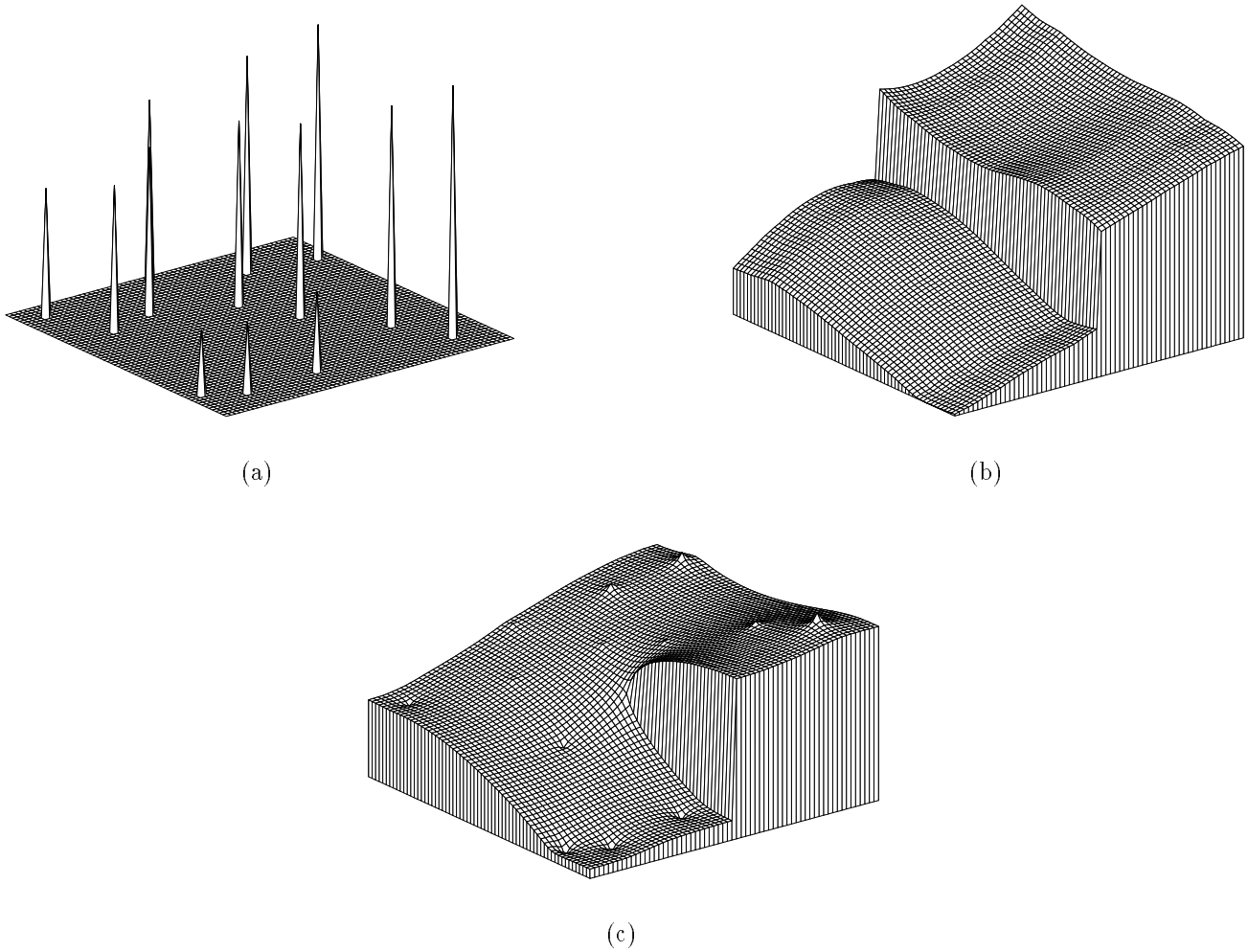

Figure 2: Surface reconstruction example; (a) original sparse data (b) solution after 1 iteration of ADI method and (c) after 10 iterations.

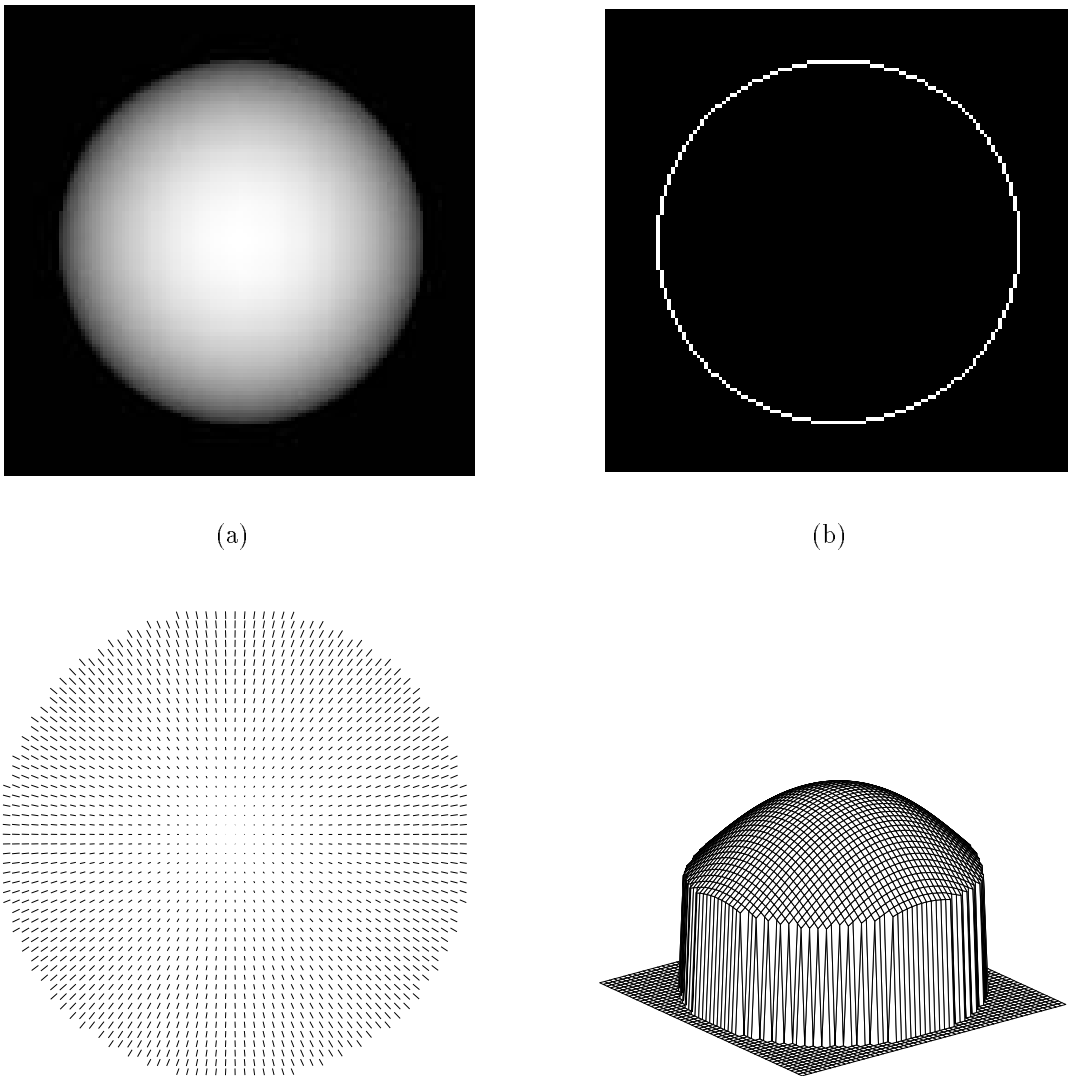

Figure 3: Shape from shading example, (a) Shaded sphere image (b) irregular occluding boundary imbedded in a rectangle (c) recovered surface orientation, (d) reconstructed surface shape.

albedo. Both depth and orientation data are specified on an occluding contour depicted in figure 3(b). Note that we have to solve three systems of equations namely $\mathbf{K}\mathbf{x}_1 = \mathbf{b}_1$, $\mathbf{K}\mathbf{x}_2 = \mathbf{b}_2$ and $\mathbf{K}_1\mathbf{x}_3 = \mathbf{b}_3$ (see section 2.1.2). The first two systems are for recovering the surface normals and the last equation for recovering the depth from the surface orientation map. We start the ADI iteration for the linear systems with the initial values of $p = q = z = 0$. The recovered surface orientation is shown in figure 3(c) and the final surface shape obtained using the orinetation information is depicted in figure 3(d). For each computed (p, q) on the RHS of equation 9, the ADI error tolerance was set to 10^{-5}. The same tolerance was used for the ADI iterations in the depth from orientation problem.

Although, we have not compared the computational time of existing solution methods for the discretized Poisson equation with the computational performance of our algorithm, we believe that our method will outperform most other existing methods due to the $O(N)$ complexity of our algorithm. Future research efforts will focus on empirically demonstrating this claim.

6 Conclusion

In this paper, we presented a novel and fast ($O(N)$) algorithm for solving the discretized Poisson equation whose solution is required in numerous low-level vision problems. We presented two major results namely, *a generalized capacitance matrix theorem and an $O(N)$ convergence result for the ADI method.* Our algorithm does not make any assumptions on the shape of the input domain unlike the polyhedral domain assumption in the convergence of multi-grid techniques [28, 29]. Thus, this makes our method the first $O(N)$ solution to the discretized Poisson equation in literature, with no assumptions on the input domain boundary.

The implication of the generalized capacitance matrix theorem presented in this paper is that, we can now apply the capacitance matrix technique to PDEs with higher order differential operators such as the biharmonic operator etc. Also, we can handle cases such as the SFS problem on an irregular domain with Neumann boundary conditions which specifically yields a singular $\mathbf{K}$ matrix. This case can not be solved using the capacitance matrix technique of Buzbee et al., [6] as mentioned in [19] because, the capacitance matrix theorem of [6] requires a nonsingular $\mathbf{K}$. Our algorithm gives

a solution to this problem (recovers a surface) within a scale factor in depth.

The techniques presented here will be useful for fast solutions to problems in other engineering fields leading to the Poisson equation. The framework of our solution holds strong promise for solving higher order PDEs in $O(N)$ time. Our future research efforts will be focussed toward extending our $O(N)$ solution to PDEs with higher order differential operators.

References

[1] B.K. Alpert. *"Wavelets and other bases for fast numerical linear algebra,"*. Academic Press Inc., 1992.

[2] G. Beylkin, R. R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithm I. *Comm. on Pure and Applied Math.*, 44:141–183, 1991.

[3] A. Blake and A. Zisserman. *Visual Reconstruction*. The MIT Press, Cambridge, MA, 1987.

[4] R.M. Bolle and B.C. Vemuri. "On three-dimensional surface reconstruction methods,". *IEEE Trans. Pattern Anal. Machine Intell.*, 13(1):1–13, 1991.

[5] Dietrich Braess. "The convergence of a multigrid method with Gauss-Seidel relaxation for the Poisson equation,". *Mathematics of Computation*, 42(166):505–519, April 1984.

[6] B. L. Buzbee, F. W. Dorr, J. A. George, and G. H. Golub. "The direct solution of the discrete Poison equation on irregular regions, ". *SIAM Journal of Numerical Analysis*, 8(4):722–736, December 1971.

[7] G. Dahlquist and A. Bjorck. *Numerical Methods*. Prentice-Hall Inc., Englewood Cliffs, NJ, 1974.

[8] R.W. Freund and N.M. Nachtigal. "QMR: a quasi-minimal residual method for non-Hermitian matrices,". Technical Report 90.51, RIACS, NASA Ames Research Center, Moffett Field, 1990.

[9] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 2nd edition, 1989.

[10] W. Hackbusch and U. Trottenberg, editors. *Multigrid Methods (Lecture Notes in Mathematics)*, volume 960. Springer-Verlag, New York, 1982.

[11] B. K. P. Horn. "Determining the lightness from an image,". *Computer Vision Graphics and Image Processing*, 3:111–299, 1974.

[12] B. K. P. Horn and M. J. Brooks, editors. *Shape from Shading.* The MIT Press, Cambridge, MA, 1989.

[13] B. K. P. Horn and B. G. Schunk. "Determining optical flow,". *Artificial Intelligence*, 17:185–203, 1981.

[14] C. Lanczos. "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators,". *J. Res. Natl. Bur. Stand.*, 45:255–282, 1950.

[15] D. Lee. "A provably convergent algorithm for shape from shading,". In *DARPA Image Understanding Workshop*, pages 489–496, Miami, December 1985. DARPA.

[16] A. Lu and E. L. Wachspress. "Solution of Lyapunov equations by alternating direction implicit iteration,". *Computers Math. Applic.*, 21(9):43–58, 1991.

[17] T. Poggio and V. Torre. "Ill-Posed problems and regularization analysis in early vision,". Technical Report AI Memo 773, MIT, A.I. LAB. Cambridge, MA,, 1984.

[18] W. Proskurowski and O. Widlund. "On the numerical solution of Helmholotz's equation by the capacitance matrix method,". *Mathematics of Computation*, 30(135):433–468, 1976.

[19] T. Simchony, R. Chellappa, and M. Shao. "Direct analytical methods for solving Poisson equations in computer vision problems,". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5), May 1990.

[20] R. Szeliski. "Fast surface interpolation using hierarchical basis functions,". *IEEE Trans. Pattern Anal. Machine Intell.*, 12(6):513–528, 1990.

[21] D. Terzopoulos. "Image analysis using multigrid relaxation methods,". *IEEE Trans. Pattern Anal. Machine Intell.*, 8:129–139, 1986.

[22] D. Terzopoulos. "regularization of inverse visual problems involving discontinuities". *IEEE Trans. Pattern Anal. Machine Intell.*, 8(4):413–424, 1986.

[23] D. Terzopoulos. "The computation of visible-surface representations,". *IEEE Trans. Pattern Anal. Machine Intell.*, 10(4):417–438, 1988.

[24] B. C. Vemuri and S. H. Lai. "A fast solution to the surface reconstruction problem,". In *SPIE conference on Mathematical Imaging: Geometric Methods in Computer Vision*, pages 27–37, San Diego, CA, July 1993. SPIE.

[25] B. C. Vemuri and R. Malladi. "Constructing intrinsic parameters with active models for invariant surface reconstruction,". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(7):668–681, July 1993.

[26] B.C. Vemuri, A. Mitiche, and J.K. Aggarwal. "Curvature-based representation of objects from range data,". *Image and Vision Comput.*, 4(2):107–114, 1986.

[27] E. L. Wachspress. "Extended application of alternating direction implicit iteration model problem theory,". *Journal of SIAM*, 11(4):994–1016, December 1963.

[28] Jinchao Xu. "Iterative methods by space decomposition and subspace correction,". *SIAM Review*, 34(4):581–613, December 1992.

[29] Xuejun Zhang. "Multilevel Schwarz methods,. *Numerische Mathematik*, 63:521–539, 1992.

Acknowledgements

This research was supported in part by the NSF grant ECS-9210648. We thank Dr. M. Shah of UCF for supplying the SFS data.