

# Physically-based Adaptive Preconditioning for Early Vision<sup>1</sup>

S. H. Lai and B. C. Vemuri

UF-CIS Technical Report TR-95-010  
Computer and Information Sciences Department  
University of Florida  
Bldg. CSE, Room 301  
Gainesville, Florida 32611-6120  
March 1995

---

<sup>1</sup>This work was supported in part by the NSF grant ECS-9210648.  
Manuscript submitted to the *IEEE Trans. on Pattern Analysis and Machine Intelligence*

## Abstract

Several problems in early vision have been formulated in the past in a regularization framework. These problems when discretized lead to large sparse linear systems. In this paper, we present a novel physically-based adaptive preconditioning technique which can be used in conjunction with a conjugate gradient algorithm to drastically improve the speed of convergence for solving the aforementioned linear systems. A preconditioner based on the membrane spline or the thin plate spline or a convex combination of the two is termed as a physically-based preconditioner for obvious reasons. The adaptation of the preconditioner to an early vision problem is achieved via the explicit use of the spectral characteristics of the regularization filter in conjunction with the data. This spectral function is used to modulate the frequency characteristics of a chosen wavelet basis leading to the construction of our preconditioner. The preconditioning technique is demonstrated for the surface reconstruction, shape from shading and optical flow computation problems. We experimentally establish the superiority of our preconditioning method over previously presented preconditioning techniques for the surface reconstruction and shape from shading problems. Performance of the preconditioning technique is demonstrated via experimental results on real and synthetic data.

# 1 Introduction

In the past decade, several problems in early vision have been formulated in a regularization framework [9, 27, 26, 30, 3, 4, 12, 13]. These formulations result in partial differential equations which when discretized lead to large sparse linear systems. Numerical iterative methods such as the Gauss-Seidal, Jacobi and the conjugate gradient technique [8] have been popular until the inception of multi-grid methods [24] and multiresolution methods [22, 19, 32]. More recently, the capacitance matrix technique [5] has been generalized to solve the linear system arising from the early vision problems very efficiently [28, 17]. However, this technique can prove to be inefficient for dense data problems with nonuniform weighting on the data constraints, such as the optical flow problem, since the size of the associated ( dense ) capacitance matrix is too large. Although this problem may be circumvented by incorporating the capacitance matrix technique as part of an iterative scheme [21], however it was pointed out in [6] that this semi-direct numerical scheme only converges when the regularization parameter  $\lambda$  is very large. *In this paper, we will introduce a physically-based adaptive preconditioning technique which when used in conjunction with the conjugate gradient algorithm, will outperform – in computational efficiency – all previously proposed preconditioning methods in literature namely, the hierarchical basis preconditioned conjugate gradient of Szeliski [22] and the change of basis methods of Yaou et al. [32] and Pentland [19].* In addition, it does not have any restriction on the choice of the regularization parameter  $\lambda$  for its convergence.

In [22, 23], Szeliski presented a preconditioning technique based on the use of hierarchical basis functions. This technique is based on the multi-level splitting of the finite element spaces presented in [33]. The hierarchical basis functions naturally lead to a pyramidal representation of the unknown function being solved for. Empirical convergence results were shown for preconditioned conjugate gradient in comparison with the standard conjugate gradient method. *A key piece of information that was overlooked in Szeliski's work [22, 23] was that, in designing a preconditioner for the conjugate gradient algorithm, no information about the problem being solved was used i.e., once a basis set was chosen, the same preconditioner was used regardless of the imposed smoothness or data constraints.*

*Thus, the true power of a preconditioning transform was not exploited to its fullest.*

More recently, Pentland [19] presented a technique for surface interpolation using a change of basis to orthonormal wavelets as a preconditioning transform. He uses the results of Beylkin et al., [2] which states that the application of  $(N \times N)$  matrices – corresponding to any pseudo-differential and Calderon-Zygmund operators – to arbitrary vectors requires either  $O(N)$  or  $O(N \log N)$  operations based on whether nonstandard or standard wavelet transform is used. The result of Beylkin et al., primarily shows a reduction in the bandwidth of the matrices corresponding to pseudo-differential operators in a wavelet basis and gives a technique whereby an  $O(N)$  coefficients can be used to approximate the  $(N \times N)$  operator, for a given tolerance. This compression takes  $O(N)$  time if the structure of the singularities of the matrix are known a priori.

There are three issues to be noted with regards to Pentland’s reported results. Firstly, after a change of basis to orthonormal wavelets, the off-diagonal terms were discarded in [19], under the claim that the transformed stiffness matrix was diagonally dominant. This diagonal dominance behavior was depicted via the profiling of a row of the transformed stiffness matrix at a high resolution. Upon close examination, we found that the *diagonal dominance* property *does not hold* in general at lower resolutions (see figure 9) and thus, discarding the off-diagonal terms is unjustified and leads to a solution which may be far from the true solution. Secondly, Pentland’s preconditioner requires the computation of the diagonal entries in a wavelet transformed stiffness matrix. Although the stiffness matrix is banded, the computation of these diagonal entries takes  $O(N \log N)$  operations, which is too expensive for large  $N$ . Thirdly, it is not known if the matrix  $(\mathbf{K} + \mathbf{S})$  in his linear system  $(\mathbf{K} + \mathbf{S})U = D$  being solved for the interpolation problem would satisfy the conditions set forth in Beylkin et al., [2] in order for their bandwidth reduction scheme to be applicable in this case. Further, we found that his algorithm fails to converge in reasonable number of iterations to the correct solution for very sparse data that is used in the examples of this paper.

Yaou and Chang [32] report a fast surface interpolation algorithm using the multiresolution wavelet transform. This work is almost exactly similar to that of Szeliski with the exception that Yaou and

Chang use a wavelet basis instead of the hierarchical basis and a Jacobi iteration [8] instead of the conjugate gradients used in Szeliski [22]. As in Szeliski, the *full potential of a preconditioning technique was not exploited* in Yaou and Chang i.e., the design of the preconditioner did not make use of any information about the problem being solved. The stiffness matrix in the surface interpolation problem was first transformed to a bi-orthonormal wavelet basis and then the transformed linear system was solved using a Jacobi iteration. Note that, in a preconditioning technique applied to the the positive definite stiffness matrix  $\mathbf{K}$ , one tries to approximate  $\mathbf{K}$  as well as possible with another positive definite matrix  $\mathbf{P}$  known as the preconditioner, such that  $\mathbf{P}^{-1}\mathbf{K}$  is a good approximation to the Identity matrix. Such an approximation  $\mathbf{P}$  was never constructed in Yaou and Chang [32]. Further, upon implementing their algorithm, it was found that their published results about rate of convergence were only valid for data from periodic function since their implementation took advantage of periodic wavelet transforms.

In this paper, we will present a very effective way to construct a physically-based adaptive preconditioner in a wavelet basis for several early vision problems namely, the surface reconstruction (SR), shape from shading (SFS) and computation of optical flow (OF). This physically-based preconditioner is adapted to the membrane spline or the thin plate spline or a convex combination of the two. Experimental results depicting the performance of our algorithm in comparison to existing methods discussed above are presented for synthetic and real data.

The rest of the paper is organized as follows. In section 2 we will very briefly present the variational formulations for the three early vision problems namely, the SR, SFS and the computation of OF, *the last of which is our new and original formulation*. Preconditioner construction in wavelet basis is presented in section 3 followed by the preconditioned conjugate gradient algorithm in section 4. Algorithm implementation on synthetic and real data are presented in section 5. In section 6, we end with a discussion and conclusion.

## 2 Variational Formulations

Variational formulations for various early vision problems have been reported in [24] and references therein. 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 [25]. The formulations involve minimization of an energy functional  $\mathcal{E}$ , which is the sum of the energy contribution from the smoothness constraint ( $\mathcal{S}$ ) and the data constraint ( $\mathcal{P}$ ) (see [25, 22] for details of this formulation).

For the surface reconstruction problem, we are faced with recovering surface shape from either sparse or dense range data. *The location of surface and orientation discontinuities (if any) are assumed to be known in the below discussion.* For a first order stabilizer, the smoothness constraint is given by

$$\mathcal{S}(v) = \int \int_{\Omega} (v_x^2 + v_y^2) \, dx dy, \quad (1)$$

where  $v(x, y)$  is the admissible function and  $v_x, v_y$  its partial derivatives assumed to be small. For the case of a second order stabilizer, we have the following smoothness constraint

$$\mathcal{S}(v) = \int \int_{\Omega} (v_{xx}^2 + 2v_{xy}^2 + v_{yy}^2) \, dx dy, \quad (2)$$

The first and second order stabilizer can be combined to form the controlled-continuity stabilizers [25]. To the smoothness constraint, we add the data constraints in the form of 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,

$$\mathcal{P}(v) = \frac{1}{2} \sum c_i (v(x_i, y_i) - d_i)^2 \quad (3)$$

Where,  $d_i$  are the depth data points specified in the domain  $\Omega$  and  $c_i$  are the uncertainty associated with the data. The goal is to find a  $u$  that minimizes the total potential energy  $\mathcal{E}(v) = \mathcal{S}(v) + \mathcal{P}(v)$ .

The numerical solution to the above problem is computed by discretizing the functionals  $\mathcal{S}(v)$  and  $\mathcal{P}(v)$  using finite element techniques [22]. The resulting energy function is a quadratic in  $\mathbf{x}$

(a discretization of  $v$ ) given by,  $E(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{K}\mathbf{x} - \mathbf{x}^T\mathbf{b} + c$ . Where,  $\mathbf{K}$  is a very large and sparse  $N \times N$  matrix [24], with  $N$  being the number of discretization nodes. The minimum  $\mathbf{x}^*$  of this energy function is found by solving the large sparse linear system  $\mathbf{K}\mathbf{x} = \mathbf{b}$ . *We present a very fast solution to this problem, based on a novel preconditioning technique, in a subsequent section.*

Numerous solution methods using variational principle formulations of the SFS problem 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 [24, 21, 11, 23]. 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 sum of the smoothness constraint and the penalty terms in this problem are,

$$\begin{aligned} \mathcal{E}(Z, p, q) &= \frac{\lambda}{2} \int \int_{\Omega} (p_x^2 + p_y^2) + (q_x^2 + q_y^2) dx dy + \frac{\mu}{2} \int \int (Z_x - p)^2 + (Z_y - q)^2 dx dy \\ &+ \int \int_{\Omega} [E(x, y) - R(p, q)]^2 dx dy. \end{aligned}$$

Where,  $E(x, y)$  is the image irradiance at a point  $(x, y)$  and  $R(p, q)$  is the relation between surface orientation  $(p, q)$  and image irradiance  $E(x, y)$ . The first term in the above equation is a weak smoothness constraint and the coefficient  $\lambda$  controls its contribution toward the final solution. The second term enforces the *integrability constraint*, ensuring that the gradients  $(p, q)$  of the recovered surface  $Z$  match those of a valid surface.  $Z_x, Z_y$  are the first partials of the surface  $Z(x, y)$ .

The above energy can be discretized directly by using finite element techniques however, the same set of discrete domain equations can be arrived at via the use of finite difference approximations for this application [23]. The discrete energy is given by

$$\mathbf{E} = \frac{1}{2} \sum (E(x, y) - R(p, q))^2 + \frac{\mu}{2} \sum (Z_x - p)^2 + (Z_y - q)^2 + \frac{\lambda}{2} \sum (p_x^2 + p_y^2 + q_x^2 + q_y^2) \quad (4)$$

where the summations are over all the pixels in the image. In vector notation, the above equation can be written as

$$\mathbf{E}(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u} - \mathbf{b}\mathbf{u} + \mathbf{C} \quad (5)$$

Where,  $\mathbf{u} = [\mathbf{z}^T \mathbf{p}^T \mathbf{q}^T]$  is a concatenation of the vector representation of the fields  $Z, p$  and  $q$ . The minimum of the above quadratic is obtained by solving  $\mathbf{K}\mathbf{u} = \mathbf{b}$  with  $\mathbf{K}\mathbf{u}$  containing linear  $\mathbf{u}$  terms and the constant and the nonlinear terms form the  $\mathbf{b}$  vector. This is an  $(N \times N)$  sparse nonlinear system which is usually solved using iterative methods. These iterative methods to solve the nonlinear system normally have the following procedures in each iteration: (1) compute the  $\mathbf{b}$  vector using the current  $\mathbf{u}$ , (2) update the solution  $\mathbf{u}$  by treating  $\mathbf{b}$  as a constant. In the past, multigrid techniques have been employed in solving this nonlinear system [24] wherein a hierarchy of problems at different resolutions needs to be explicitly constructed. In [23], Szeliski presented a fast shape from shading algorithm which employs hierarchical basis for preconditioning the conjugate gradient algorithm used to solve the above nonlinear system. Unlike the multigrid methods, this technique does not require the explicit construction of a multiresolution pyramid of problems. *In this paper we present a novel preconditioning method that leads to a faster convergence than the one in [23] and uses a wavelet basis.* Our method makes use of the spectral characteristics of the data and the imposed smoothness constraint. This spectral function is then used to modulate the frequency characteristics of the chosen wavelet basis leading to the construction of the preconditioner.

A third early vision problem that we consider in this paper is the computation of optical flow from an image sequence. Recently, Barron et al., [1] have presented a comprehensive survey of various optical flow computation techniques along with a quantitative comparison of them. *We note that the survey did not do any comparison of the computation time required by the various algorithms. In this paper, we will develop a novel preconditioning technique applicable to any gradient based optical flow computation technique that uses a smoothness constraint involving the 2D Laplacian operator. We develop a new formulation which modifies the Horn and Schunk formulation [13] to incorporate a separate condition along discontinuities in the image intensity [10]. This modification drastically enhances the performance of our method over other gradient-based methods because the optical flow constraint equation is invalid along intensity discontinuities and causes large errors in the computed flow. We get drastically better and more robust estimates of optical flow than any gradient based optical flow technique reported in Barron et al., [1].* Using wavelet basis, we develop a preconditioner

for this problem which assists our algorithm in converging very fast.

Our variational formulation of the optical flow problem involves minimizing a combination of the optical flow constraint equation with a global smoothness term and an additional constraint term along edges leading to

$$\int_D \alpha((\nabla E \bullet \mathbf{u}) + E_t)^2 + \lambda(\|\nabla u\|_2^2 + \|\nabla v\|_2^2) d\mathbf{x} + \oint \beta((\nabla(\nabla^2 G * E) \bullet \mathbf{u}) + (\nabla^2 G * E)_t)^2 d\mathbf{x} \quad (6)$$

Where  $\mathbf{u}(\mathbf{x}, t) = (u(\mathbf{x}, t), v(\mathbf{x}, t))$  is the velocity field to be estimated,  $E$  is the image brightness function, and  $\nabla^2 G$  is the Laplacian of Gaussian operator [10]. The first term is the gradient constraint which is active away from discontinuities, the second term is the smoothness constraint on the velocity field with  $\lambda$  controlling the contribution of the smoothness term and the third term is the flow constraint along edges which is disabled away from the edges. A discrete version of the above energy can be written as

$$\sum (\overline{E}_x \overline{u} + \overline{E}_y \overline{v} + \overline{E}_t)^2 + \frac{\lambda}{2} \sum (\overline{u}_x^2 + \overline{u}_y^2) + (\overline{v}_x^2 + \overline{v}_y^2) \quad (7)$$

Where,  $\overline{u}$  and  $\overline{v}$  form the discretized flow vector,  $\overline{u}_x, \overline{u}_y, \overline{v}_x$  and  $\overline{v}_y$  represent the discretized partials of the velocity field,  $\overline{E}_x, \overline{E}_y$  and  $\overline{E}_t$  correspond to the discretized partials in the  $x, y$  and  $t$  directions respectively – collectively obtained from the first and the third terms of the equation 6. Note that  $\overline{E}_x, \overline{E}_y$  and  $\overline{E}_t$  are the discretized partials of the brightness function  $E$  when the location of differentiation is away from discontinuities, and they represent the discretized partials of  $\nabla^2 G * E$  when the differentials are taken at the discontinuity locations. The minimization of this discretized energy leads to solving a linear system of equations  $\mathbf{K}\mathbf{u} = \mathbf{b}$  where  $\mathbf{K}$  contains the discretized differential (smoothness) operator along with  $\overline{E}_x, \overline{E}_y$  and the  $\mathbf{b}$  contains the terms corresponding to the optical flow constraint equation. The gradient constraint in the above energy function implies the weighting  $(\overline{E}_x^2 + \overline{E}_y^2)$  on  $(\overline{u}, \overline{v})$  at each discretized spatial location. This weighting is not reasonable since the gradient constraints at locations of high gradient are not necessarily reliable. In fact, they tend to be unreliable because the brightness gradients close to discontinuities are normally higher than those away from discontinuities. Therefore, we eliminate this weighting for each gradient constraint via normalization. To have a uniform contribution at each pixel from the flow constraint, we normalize

the entries  $\overline{E}_x^2, \overline{E}_y^2, \overline{E}_x \overline{E}_y$  in the matrix  $\mathbf{K}$  with  $\overline{E}_x + \overline{E}_y$ .

### 3 Adaptive Preconditioning in a Wavelet Basis

Iterative methods are preferable to direct methods for solving large and sparse linear systems in early vision problems, since direct methods require the formation of intermediate matrices which are not necessarily sparse. Although the sparsity can be preserved when using direct methods on 1D problems with the smoothness constraint, it will be destroyed in the case of 2D problems. This makes the number of operations and the amount of storage too large for 2D and higher dimensional problems.

The efficiency of applying an iterative method to solve a linear system depends on its rate of convergence, which is a function of the condition number of the coefficient matrix in a linear system. Unfortunately, many early vision problems after regularization lead to large ill-conditioned linear systems. Therefore, most iterative methods converge very slowly to the true solution and are thus computationally impractical. One way to speed up the convergence of an iterative method is via preconditioning. In this section, we present a new adaptive preconditioning technique wherein the preconditioner is constructed in a wavelet basis. Since the proposed preconditioner is designed to adapt to the spectral characteristics of the smoothness constraint in the regularization solution (in the frequency domain) of different early vision problems, it is bound to perform better than the existing “fixed” preconditioners.

#### 3.1 Preconditioning Technique

For a linear system  $\mathbf{K}\mathbf{x} = \mathbf{b}$  with positive-definite matrix  $\mathbf{K}$ , we assume there exists a positive-definite matrix  $\mathbf{P}$ , the preconditioner for the matrix  $\mathbf{K}$ , such that the condition number of  $\mathbf{P}^{-1}\mathbf{K}$  is greatly reduced. Since  $\mathbf{P}$  is a positive-definite matrix, it can be shown that there exists a positive-definite matrix  $\mathbf{C}$  such that  $\mathbf{C}^2 = \mathbf{P}$  [8]. Now, we can define the vectors  $\tilde{\mathbf{x}}$  and  $\tilde{\mathbf{b}}$  via the transformation

matrix  $\mathbf{C}$  as follows

$$\tilde{\mathbf{x}} = \mathbf{C}\mathbf{x}, \quad \tilde{\mathbf{b}} = \mathbf{C}^{-1}\mathbf{b}. \quad (8)$$

Thus, the linear system  $\mathbf{K}\mathbf{x} = \mathbf{b}$  can be solved by solving the transformed linear system

$$\tilde{\mathbf{K}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}, \quad (9)$$

where  $\tilde{\mathbf{K}} = \mathbf{C}^{-1}\mathbf{K}\mathbf{C}^{-1}$ . Note that the matrix  $\tilde{\mathbf{K}}$  in the transformed linear system is *similar* to the matrix  $\mathbf{P}^{-1}\mathbf{K}$ , i.e., the condition numbers of  $\tilde{\mathbf{K}}$  and  $\mathbf{P}^{-1}\mathbf{K}$  are equivalent. The condition number of the transformed matrix is much smaller than that of the original matrix according to the aforementioned criterion of choosing a preconditioner  $\mathbf{P}$ . Consequently, the convergence of the iterative methods can be accelerated by a transformation of the original linear system to a better conditioned linear system.

Although the conditioning of a linear system can be improved after an appropriate transformation, it is *not practical to form the matrix  $\tilde{\mathbf{K}}$*  to solve equation 9 since, the transformation of a matrix requires a large number of operations and the sparsity structure of the matrix  $\mathbf{K}$  could be destroyed after the transformation. Fortunately, it is not necessary to compute this transformation of the matrix  $\mathbf{K}$  to achieve the preconditioning effect in a conjugate gradient algorithm. Instead, the same preconditioning can be achieved via solving an auxiliary linear system  $\mathbf{P}\mathbf{z} = \mathbf{r}$ , where  $\mathbf{r}$  is the residual vector, in each conjugate gradient iteration. (The preconditioned conjugate gradient algorithm is given in section 4.) Therefore, the *second consideration for a good preconditioner is to choose the matrix  $\mathbf{P}$  such that the linear system  $\mathbf{P}\mathbf{z} = \mathbf{r}$  can be easily solved.*

The above two criteria for designing a good preconditioner in conjugate gradient are equally important. Reducing the condition number can accelerate the convergence of the iterative methods employed. Choosing the preconditioner  $\mathbf{P}$  such that  $\mathbf{P}\mathbf{z} = \mathbf{r}$  is easily solvable is very crucial for efficiently updating the solution in each iteration of the preconditioned conjugate gradient (PCG) algorithm. Our preconditioner is designed in a wavelet basis by approximating the spectral characteristics of the matrix  $\mathbf{K}$ . This approximation is achieved via the frequency domain analysis of the regularization operators and using the frequency band partitioning behavior of a wavelet basis. Therefore, our adaptive preconditioner is akin to a spectrally equivalent operator for the original

matrix  $\mathbf{K}$ . Furthermore, the solution of a linear system using our adaptive preconditioner can be obtained by taking the wavelet transform, scaling the coefficients, and taking the inverse wavelet transform. By using a very compactly supported wavelet, the solution update can be achieved very efficiently.

### 3.2 Regularization Filter

The filtering interpretation of regularization operators has been discussed in vision literature [25, 27]. The inverse of the control-continuity stabilizer (Laplace + Bi-harmonic operators) may be interpreted as a low-pass filter hence, the solution to an early vision problem in a regularization framework can be obtained by applying this low-pass filter to the data. We will use this filtering behavior of the regularization operator to develop a preconditioner which approximates this filter in a wavelet basis.

In section 2, we have seen the variational formulations of various early vision problems that lead to minimizing an energy functional. This minimization can be achieved by solving the associated Euler-Lagrange equation which for the surface reconstruction problem in the continuous data case (with a controlled-continuity stabilizer) can be written as

$$\rho(\mathbf{x})\{(1 - \tau(\mathbf{x}))\Delta^2 v(\mathbf{x}) - \tau(\mathbf{x})\Delta v(\mathbf{x})\} + c(\mathbf{x})v(\mathbf{x}) = c(\mathbf{x})d(\mathbf{x}), \quad (10)$$

where  $\Delta$  is the Laplace operator,  $\Delta^2$  is the bi-harmonic operator,  $c(\mathbf{x})$  is the weighting function associated with the data  $d(\mathbf{x})$ , the functions  $\rho(\mathbf{x})$  and  $\tau(\mathbf{x})$  control the rigidity and tension of the solution surface respectively. Now let's consider the case when  $\rho(\mathbf{x})$  and  $\tau(\mathbf{x})$  are constant functions, i.e.  $\rho(\mathbf{x}) = \rho$  and  $\tau(\mathbf{x}) = \tau$ . In addition, let's also assume the function  $c(\mathbf{x})$  to be a constant function, which means the Gaussian noise associated with each data measurement has the same variance. Taking the Fourier transform of equation 10 we get

$$\mathcal{V}(\boldsymbol{\omega}) = \left( \frac{c}{\rho(1 - \tau)\|\boldsymbol{\omega}\|^4 + \rho\tau\|\boldsymbol{\omega}\|^2 + c} \right) \mathcal{D}(\boldsymbol{\omega}). \quad (11)$$

Where  $\boldsymbol{\omega} = (\omega_1, \dots, \omega_d)$  is the d-dimensional frequency domain variable while,  $\mathcal{V}(\boldsymbol{\omega})$  and  $\mathcal{D}(\boldsymbol{\omega})$  are the Fourier transforms of  $v(\mathbf{x})$  and  $d(\mathbf{x})$  respectively. Equation 11 can be given a filtering interpretation

namely,  $\mathcal{V}(\boldsymbol{\omega})$  is obtained by low-pass filtering the data  $\mathcal{D}(\boldsymbol{\omega})$ . This low-pass filter is characterized by the first term on the right hand side in equation 11 called the transfer function. *Our preconditioner is constructed in a wavelet basis to approximate this transfer function.*

Equation 11 expresses a linear shift-invariant low-pass filtering of the data in frequency domain, derived from a regularization framework. This filtering interpretation can be generalized to the regularly sampled discrete data case in a straightforward manner. However, this linear shift-invariant filter will become a nonlinear and spatially varying filter in the general cases when the functions  $\rho(\mathbf{x})$ ,  $\tau(\mathbf{x})$  and  $c(\mathbf{x})$  are not constant. These cases include discontinuity-preserving regularization and sparse-data interpolation. Although the corresponding nonlinear and spatially varying filter is difficult to characterize fully in the frequency domain, it is in general a low-pass filter and we can approximate its spectral characteristics in the frequency domain. In the next section we show how to achieve this approximation in a wavelet basis.

### 3.3 Wavelet Basis

Wavelet transform has been used widely in the signal and image processing communities since it provides a powerful tool for multiresolution analysis. The construction of a wavelet basis is achieved by defining a scaling function  $\phi(x)$ , such that the integer translates of the dilated scaling function  $\sqrt{2^{-j}}\phi(2^{-j}x)$  form an orthonormal basis of  $L^2(\mathbb{R})$ , where  $j \in \mathbf{Z}$ . This orthonormal basis ( $= \{\sqrt{2^{-j}}\phi(2^{-j}x - k)\}_{k \in \mathbf{Z}}$ ) forms a subspace  $\mathbf{V}_j \subset \mathbf{L}^2(\mathbb{R})$ , and the subspaces  $\mathbf{V}_j$  satisfy the containment hierarchy

$$\mathbf{V}_j \subset \mathbf{V}_{j-1}, \forall j \in \mathbf{Z}. \quad (12)$$

Now we can introduce the difference space  $\mathbf{W}_j$  between the two consecutive subspaces  $\mathbf{V}_j$  and  $\mathbf{V}_{j-1}$  such that  $\mathbf{V}_j \oplus \mathbf{W}_j = \mathbf{V}_{j-1}$ , where  $\oplus$  denotes the direct sum operator. The key idea of the wavelet basis is the difference space  $\mathbf{W}_j$  is spanned by an orthonormal basis which are obtained via integer translations of the associated wavelet function  $\psi(x)$ , i.e.  $\mathbf{W}_j = \{\sqrt{2^{-j}}\psi(2^{-j}x - k)\}_{k \in \mathbf{Z}}$ . Therefore, a

multiresolution orthogonal decomposition of the space  $\mathbf{V}_0$  can be achieved via

$$\mathbf{V}_J \oplus \mathbf{W}_J \oplus \mathbf{W}_{J-1} \oplus \cdots \oplus \mathbf{W}_1 = \mathbf{V}_0, \quad (13)$$

where  $J$  is a positive integer. From this multiresolution orthogonal space decomposition, we can construct a multiresolution orthonormal wavelet basis, since each of the subspaces on the left hand side of equation 13 are orthogonal to each other and there exists an orthonormal basis for each subspace as discussed above.

The wavelet decomposition and reconstruction can be efficiently achieved via a QMF (Quadrature Mirror Filters) implementation as shown in figure 1 (see [18]). In this figure,  $\mathbf{A}_j \mathbf{f}$  is the discrete approximation of the function  $\mathbf{f}$  in the subspace  $\mathbf{V}_j$ , i.e.  $\mathbf{A}_j \mathbf{f}$  contains the values of the inner products of the function  $\mathbf{f}$  and the orthonormal basis functions in  $\mathbf{V}_j$ , and  $\mathbf{D}_j \mathbf{f}$  encodes the projection of the function  $\mathbf{f}$  on the subspace  $\mathbf{W}_j$ . The filters  $\tilde{\mathbf{H}}$  ( in decomposition) and  $\mathbf{H}$  ( in reconstruction) are low-pass filters, while the filters  $\tilde{\mathbf{G}}$  ( in decomposition) and  $\mathbf{G}$  ( in reconstruction) are high-pass filters. They are usually designed to be the FIR filters, which can be implemented via convolution operations. With an appropriate choice of a wavelet basis, we can find the associated FIR filters with very short length/span and the QMF filter implementation for the wavelet transform can be accomplished very efficiently.

The QMF structure decomposes the space from one level to the next coarser level with each stage involving a low-pass and high-pass filtering. From a frequency domain analysis, we can see that a wavelet decomposition divides the frequency domain into two separate bands in each stage. Therefore, each subspace  $\mathbf{W}_j$  has its own frequency band, which means that all the wavelet basis functions in any particular level are fully contained in the frequency band associated with that level. This behavior of dividing the frequency domain [20, 29] exhibited by a wavelet decomposition is illustrated in figure 2. Notice that the closer the QMF filters are to the ideal low-pass and high-pass filters, the better it is for achieving a good frequency-domain division, i.e., the overlap between the frequency bands associated with each level is minimized with close approximations of the ideal low and high-pass filters.

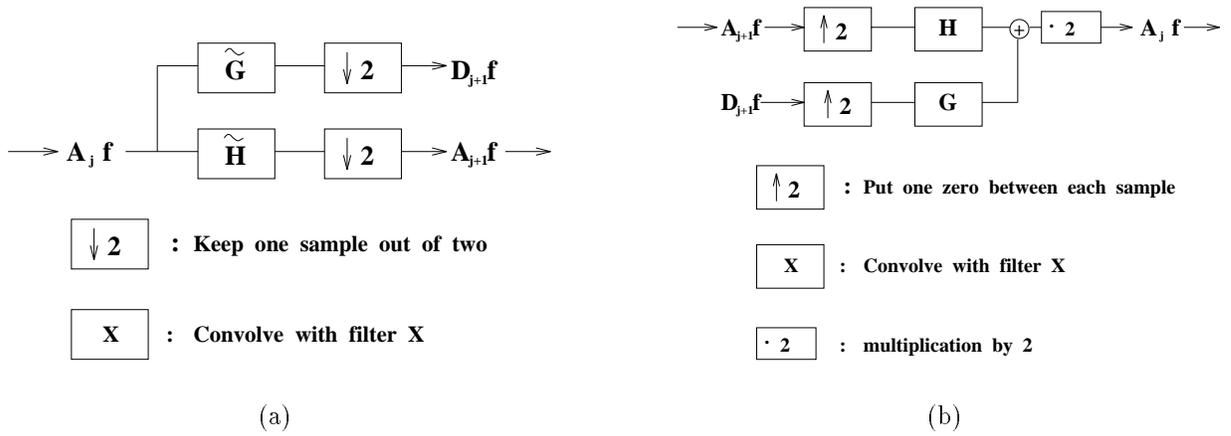


Figure 1: The implementation of the wavelet transform via the QMF structure from one level to the next, (a) decomposition (b) reconstruction, adopted from Mallat[1989].

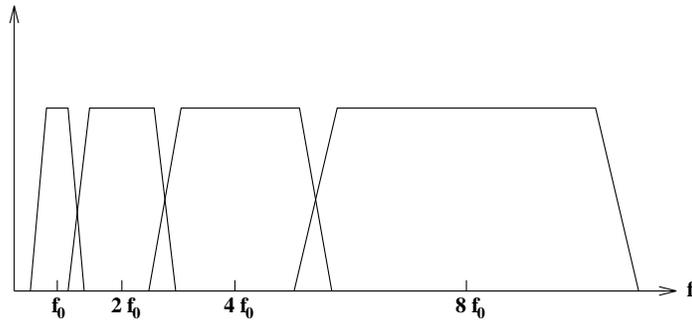


Figure 2: Division of the frequency domain in a wavelet basis.

In [18], Mallat constructed the 2D wavelet basis by using the tensor products of the 1D wavelet and scaling functions. The QMF implementation of the wavelet transform was also extended to the two-dimensional case in [18]. The division of frequency domain behavior for the 2D wavelet basis is simply an extension of the 1D case.

After introducing the division of frequency domain behavior for a wavelet basis, we are ready to construct our preconditioner in a wavelet basis and adapt its spectrum to that of the regularization filter discussed in section 3.2.

### 3.4 Adaptive Preconditioning via the Modulation in a Wavelet Basis

In section 3.1, we discussed two criteria for selecting a good preconditioner. In this section, we construct an adaptive preconditioner for early vision problems by following these two criteria.

Using techniques of frequency domain analysis, a low-pass filtering interpretation can be given to the regularization formulation of early vision problems as was shown in section 3.2. The spectral characteristics of the regularization filter in the frequency domain can be approximated using the frequency domain subdivision behavior of the wavelet transform which can be implemented very efficiently using a quadrature mirror based implementation.

To apply a preconditioner  $\mathbf{P}$  to a matrix  $\mathbf{K}$ , the *first criterion* is to drastically reduce the condition number of  $\mathbf{P}^{-1}\mathbf{K}$ ; the *second criterion* for a good preconditioner is that the auxiliary linear system  $\mathbf{P}\mathbf{z} = \mathbf{r}$  should be efficiently solvable. An obvious choice to reduce the condition number of  $\mathbf{P}^{-1}\mathbf{K}$  is to choose  $\mathbf{P} = \mathbf{K}$ , which reduces the condition number to 1. Although only one iteration suffices to solve the problem, the complexity of solving the auxiliary linear system  $\mathbf{P}\mathbf{z} = \mathbf{r}$  is the same as that of solving the original linear system. Therefore, nothing is gained by using  $\mathbf{P} = \mathbf{K}$ . A reasonable choice for a good preconditioner is to find a matrix  $\mathbf{P}$  that is spectrally close to  $\mathbf{K}$ ; in addition, the solution to the auxiliary linear system  $\mathbf{P}\mathbf{z} = \mathbf{r}$  should be easily solvable. *Our preconditioner is constructed based on this very idea.*

The matrix  $\mathbf{K}$  is the stiffness matrix obtained from a discretization of the regularization formulation. Therefore, its inverse has the same spectral characteristics as that of the regularization filter given in equation 11. Now we approximate the spectral characteristics of the regularization filter in a wavelet basis by using its frequency domain partitioning property. *This approximation is accomplished by modulating each wavelet basis function by the spectral value of the regularization filter taken at the center of the corresponding frequency domain subregion.* To be more specific, when Mallat's tensor product strategy [18] is used for the 2D wavelet transform, each level (resolution) contains three sets of wavelet basis functions, i.e. LH ( $\phi \otimes \psi$ ), HL ( $\psi \otimes \phi$ ), and HH ( $\psi \otimes \psi$ ), and each set corresponds

to its own subregion in the 2D frequency domain. *The modulation for the wavelet basis functions in each set is based on the spectral function of the regularization filter in its corresponding subregion.* For ease of implementation, we can just take the spectral function value at the center of the subregion to be the modulation factor.

The inverse of the stiffness matrix is basically a low-pass filter, whose spectral function changes rapidly in the lower frequency region and varies smoothly in the higher frequency region. An approximation to the spectral function of the low-pass filter achieved in a wavelet basis is very fine in the lower frequency regions and coarse in the higher frequency regions. This is because the division of the frequency domain becomes finer in the lower-frequency band with increasing levels (see figure 2). Therefore, we can achieve a better approximation when more levels are used. In this paper, the number of levels used in the preconditioner is determined by the support of the wavelet basis used and the problem size. *Therefore, for a given problem size and wavelet basis, the number of levels is fixed.*

Our preconditioner  $\mathbf{P}$  can be written as

$$\mathbf{P} = \mathbf{W}\mathbf{M}\mathbf{W}^T, \tag{14}$$

where the matrix  $\mathbf{W}$  is a 2D orthonormal wavelet transformation matrix and the matrix  $\mathbf{M}$  is the diagonal modulation matrix, whose diagonal values are obtained from the spectral function of the regularization filter. From equation 14, we can see that the solution to the auxiliary linear system  $\mathbf{P}\mathbf{z} = \mathbf{r}$  in the preconditioned conjugate gradient can be computed very efficiently when the wavelet basis are chosen appropriately, i.e. the QMF filters are of short length/span. As long as the QMF filters are of finite length, the solution can be obtained in  $O(N)$  operations. The solution to this auxiliary linear system is obtained in the three steps namely, (1) take the wavelet transform of the vector  $\mathbf{r}$ , (2) modulate the transformed vector using the matrix  $\mathbf{M}$ , and (3) take the inverse wavelet transform.

## 4 Adaptive Preconditioned Conjugate Gradient Algorithm

As discussed in section 2, the discretization of the variational formulation leads to solving the system of equations  $\mathbf{K}\mathbf{x} = \mathbf{b}$ . The stiffness matrix  $\mathbf{K}$  is symmetric positive definite and therefore, the conjugate gradient algorithm can be used to solve the problem guaranteeing a unique solution. When a preconditioner  $\mathbf{P}$  is used to accelerate the convergence, we have what is known as the preconditioned conjugate gradient algorithm (see [8]) given as follows.

0. Choose an initial  $\mathbf{x}_0$ , and compute  $\mathbf{r}_0 = \mathbf{b} - \mathbf{K}\mathbf{x}_0$ ; set  $k = 0$ .
1. Solve  $\mathbf{P}\mathbf{z}_k = \mathbf{r}_k$ ;  $k = k + 1$ .
2. If  $k = 1$ , set  $\mathbf{p}_1 = \mathbf{z}_0$ ; else compute  $\beta_k^D = \mathbf{r}_{k-2}^T \mathbf{z}_{k-2}$  and  $\beta_k = \alpha_{k-1}^N / \beta_k^D$ ;  
update  $\mathbf{p}_k = \mathbf{z}_{k-1} + \beta_k \mathbf{p}_{k-1}$ .
3. Compute  $\alpha_k^N = \mathbf{r}_{k-1}^T \mathbf{z}_{k-1}$ ,  $\alpha_k^D = \mathbf{p}_k^T \mathbf{K} \mathbf{p}_k$ , and  $\alpha_k = \alpha_k^N / \alpha_k^D$ ;
4. Update  $\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k \mathbf{p}_k$ .
5. Update  $\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_k \mathbf{p}_k$ .
6. If  $\mathbf{r}_k \simeq \mathbf{0}$ , stop; else go to step 1.

The only difference between the preconditioned conjugate gradient algorithm and the standard conjugate gradient algorithm is the preconditioning process in step 1. Since the preconditioning involves solving the auxiliary linear system  $\mathbf{P}\mathbf{z}_k = \mathbf{r}_k$  in each iteration, it is absolutely necessary to have a fast solution to this preconditioning equation when designing a good preconditioner.

The strategy for constructing our adaptive preconditioner for the early vision problems with regularization formulation was discussed in section 3. Our preconditioner is based on the idea of modulation in a wavelet basis to approximate the spectral characteristics of the matrix  $\mathbf{K}$ . Accordingly, our adaptive wavelet preconditioner  $\mathbf{P}$  took the form  $\mathbf{W}\mathbf{M}\mathbf{W}^T$  with the diagonal modulation matrix  $\mathbf{M}$

encoding the spectral approximation in the wavelet basis. Now, the primary problem in this preconditioner construction is *how to construct the modulation matrix  $\mathbf{M}$  for general problems. In the following paragraphs, we show how to achieve the construction of an adaptive preconditioner for the surface reconstruction, shape from shading, and optical flow computation problems.*

## 4.1 Surface Reconstruction Problem

For the surface reconstruction, the stiffness matrix  $\mathbf{K}(= \mathbf{K}_d + \mathbf{K}_s)$  contains the matrix  $\mathbf{K}_d$  obtained from the data constraint and the matrix  $\mathbf{K}_s$  corresponding to the smoothness constraint. When the controlled-continuity stabilizer is used, the matrix  $\mathbf{K}_s$  is the linear combination of the discrete Laplacian matrix ( from membrane) and the discrete biharmonic matrix ( from thin plate). Since the spectral distributions for both matrices are known a priori, the spectral distribution for the matrix  $\mathbf{K}_s$  can be obtained from their linear combination. Now, what remains to be determined is the spectral distribution for the matrix  $\mathbf{K}_d$ . For the dense data case with uniform weights for each data constraint, the matrix  $\mathbf{K}_d$  is a scaled identity matrix, whose spectral distribution is a constant function. This case corresponds to the discrete version of the linear shift-invariant low-pass filtering interpretation for the regularization problem. For other cases such as sparse data surface reconstruction or dense data with nonuniform weights, we somehow need to estimate the spectral distribution of  $\mathbf{K}_d$  in the frequency domain. We will elaborate further on this spectral estimation in the following paragraphs.

Since the spectral approximation of the matrix  $\mathbf{K}$  is achieved in the wavelet basis, we can approximate the spectral distribution of the matrix  $\mathbf{K}_d$  via the wavelet transform. One way is to do the wavelet transform of the matrix  $\mathbf{K}_d$  and then compute the energy contained in each diagonal block which corresponds to a particular division of the frequency band, and the computed energy is used to represent the spectral value for this frequency band. To use this spectral estimation, we need to develop a fast computational algorithm since the wavelet transform of an  $N \times N$  matrix requires  $O(N^2)$  operations. *Another way to estimate the spectral distribution of  $\mathbf{K}_d$  is to take the wavelet transform of its diagonal entries only. The  $N$  diagonal entries of the  $N \times N$  diagonal matrix  $\mathbf{K}_d$*

are grouped into an  $n \times n$  matrix. Then we compute the energy from the wavelet transform of this matrix and associate it with the spectral value of each frequency band. This method takes advantage of the diagonal structure of the matrix  $\mathbf{K}_d$  and can be accomplished in  $O(N)$  operations. We use this method to estimate the spectrum of the matrix  $\mathbf{K}_d$  in our experiment.

Using the above discussed spectral approximation to the matrices  $\mathbf{K}_s$  and  $\mathbf{K}_d$  separately in a wavelet basis, we can obtain the diagonal approximation matrices  $\mathbf{M}_s$  and  $\mathbf{M}_d$ , respectively. Then, the modulation matrix  $\mathbf{M}$  is taken to be the sum of  $\mathbf{M}_s$  and  $\mathbf{M}_d$ . Since the matrix  $\mathbf{M}_d$  is obtained from the spectral approximation and the matrix  $\mathbf{M}_s$  has very small diagonal entries at the very coarse resolution levels, the spectral estimation error in  $\mathbf{M}_d$  at these levels may cause significant error in the spectral approximation to the matrix  $\mathbf{K}$ . This problem may exist only in the case of sparse data surface reconstruction, since the spectral estimation for  $\mathbf{K}_d$  in the dense data case is quite accurate.

To amend the problem caused by the spectral estimation error, we introduce a weighting function  $\sigma(i)$ , where  $i$  is the number of iterations, to appropriately weight the importance of  $\mathbf{M}_s$  and  $\mathbf{M}_d$  during the CG iterations, i.e.,

$$\mathbf{M}(i) = \mathbf{M}_s + \sigma(i)\mathbf{M}_d. \quad (15)$$

The matrix  $\mathbf{M}(i)$  is the modulation matrix in the  $i$ -th iteration. From the filtering correspondence with the regularization formulation, we can see that the overall spectral characteristics for the matrix  $\mathbf{K}^{-1}$  is a low-pass filter. Note that the inverse of  $\mathbf{K}_s$  is also a low-pass filter, while the inverse of  $\mathbf{K}_d$  is not. Therefore, the matrix  $\mathbf{M}_s$  ( or  $\mathbf{K}_s$  ) is more dominant than the matrix  $\mathbf{M}_d$  ( or  $\mathbf{K}_d$  ) for solving the linear system  $\mathbf{K}\mathbf{x} = \mathbf{b}$ . Based on this observation, we set the weighting function  $\sigma(i)$  to be very small initially and then gradually increase the values of the weighting function. Using this schedule for the weighting function, the lower frequency components of the solution are recovered at the beginning of the preconditioned conjugate gradient iterations, and then the higher frequency components. This global-to-local shape recovery phenomenon can be predicted by the choice of the weighting function  $\sigma(i)$ . In fact, the weighting function can be adapted during the iterations instead of following a preset schedule. The adaptation concept is similar to the adaptive step scheme in the

scale space tracking [31].

The preconditioner construction for the shape-from-shading and the optical flow computation problems is very similar to the one outlined here for the surface reconstruction problem. We only state their differences in the following paragraphs.

## 4.2 Shape from Shading Problem

The variational formulation of the SFS problem leads to solving a set of nonlinear partial differential equations. After discretization, we need to solve the system  $\mathbf{K}\mathbf{x} = \mathbf{b}$ . If the linear approximation of the reflectance function technique [11] is used to improve the convergence, the stiffness matrix  $\mathbf{K} \in \mathfrak{R}^{3N \times 3N}$  has the following form.

$$\mathbf{K} = \begin{bmatrix} \mu\mathbf{K}_s & \mu\mathbf{I} \otimes \mathbf{D} & \mu\mathbf{D} \otimes \mathbf{I} \\ \mu\mathbf{I} \otimes \mathbf{D}^T & \lambda\mathbf{K}_s + (R_p + \mu)\mathbf{I} & R_p R_q \mathbf{I} \\ \mu\mathbf{D}^T \otimes \mathbf{I} & R_p R_q \mathbf{I} & \lambda\mathbf{K}_s + (R_q + \mu)\mathbf{I} \end{bmatrix} \quad (16)$$

where  $\mathbf{K}_s \in \mathfrak{R}^{N \times N}$  is the Laplacian matrix,  $\mathbf{I} \in \mathfrak{R}^{N \times N}$  is the identity matrix,  $R_p$  and  $R_q$  are the first-order partial derivatives of the reflectance function with respect to  $p$  and  $q$ , respectively, taken at the reference gradient  $(p_0, q_0)$ , and the matrix  $\mathbf{D} \in \mathfrak{R}^{n \times n}$  ( $N = n^2$ ) has nonzero entries only along the main diagonal and upper diagonal with 1s along the main diagonal and  $-1$ s along the upper diagonal, i.e.,

$$\mathbf{D} = \begin{bmatrix} 1 & -1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & -1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & 1 & -1 \\ 0 & \cdots & \cdots & \cdots & 0 & 0 \end{bmatrix}. \quad (17)$$

Our preconditioner  $\mathbf{P}$  has similar block structure as  $\mathbf{K}$  and can be expressed as

$$\mathbf{P} = \begin{bmatrix} \mathbf{W} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{W} \end{bmatrix} \begin{bmatrix} \mathbf{M}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_3 \end{bmatrix} \begin{bmatrix} \mathbf{W}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{W}^T \end{bmatrix}. \quad (18)$$

The modulation matrices  $\mathbf{M}_1$ ,  $\mathbf{M}_2$  and  $\mathbf{M}_3$  are all diagonal and they are used to modulate the wavelet basis functions corresponding to  $\mathbf{z}$ ,  $\mathbf{p}$ , and  $\mathbf{q}$  vectors, respectively. The modulation matrices

are constructed to approximate the spectral distributions of the corresponding diagonal blocks in the matrix  $\mathbf{K}$ , i.e.  $\mu\mathbf{K}_s$ ,  $\lambda\mathbf{K}_s + (R_p + \mu)\mathbf{I}$  and  $\lambda\mathbf{K}_s + (R_q + \mu)\mathbf{I}$ . The construction procedure for each block is the same as discussed in the surface reconstruction problem.

### 4.3 Optical Flow Computation Problem

Similar to the SFS problem, the stiffness matrix  $\mathbf{K}$  for the optical flow estimation has the following  $2 \times 2$  block structure.

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_s + \overline{\mathbf{E}}_{xx} & \overline{\mathbf{E}}_{xy} \\ \overline{\mathbf{E}}_{xy} & \mathbf{K}_s + \overline{\mathbf{E}}_{yy} \end{bmatrix}, \quad (19)$$

where  $\overline{\mathbf{E}}_{xx}$ ,  $\overline{\mathbf{E}}_{xy}$ , and  $\overline{\mathbf{E}}_{yy}$  are all diagonal matrices with the normalized entries  $\frac{\overline{E}_x^2}{\overline{E}_x^2 + \overline{E}_y^2}$ ,  $\frac{\overline{E}_x \overline{E}_y}{\overline{E}_x^2 + \overline{E}_y^2}$ , and  $\frac{\overline{E}_y^2}{\overline{E}_x^2 + \overline{E}_y^2}$ . The construction of the block preconditioner is similar to that in the SFS problem, therefore it is omitted here.

## 5 Experimental Results

In this section, we present implementation results of applying our adaptive preconditioner to the surface reconstruction, shape-from-shading, and optical flow computation problems. Since our preconditioner adapts to the problem, it is bound to perform better than the existing fixed preconditioners described in literature. We will justify this claim through the experimental results.

Note that there is no restriction on the wavelet basis being used in the construction of our preconditioner. However, the efficiency of our preconditioner is related to the localization of the wavelet being used. The time localization makes the computation of the wavelet transform efficient; the localization in the frequency domain yields a more accurate spectral approximation and therefore faster convergence of the preconditioned conjugate gradient can be achieved. In our experiment, we use the cubic B-spline wavelet given in [18] with the QMF filters truncated to a length of 11. This truncation is used to reduce the operations involved in QMF implementation of the wavelet transform and an insignificant error is introduced in the frequency domain.

In constructing our preconditioner, the weighting function  $\sigma(i)$  is needed to control the contributions of  $\mathbf{K}_s$  and  $\mathbf{K}_d$  to the modulation matrix. We suggest that the weighting function be chosen as an increasing function. We used an increasing function of the form  $(1 - e^{-\frac{i}{c}})^d$ , where  $c$  and  $d$  are constants, for the weighting function. In our experiment,  $c$  was usually set to 10 (for problems not severely ill-conditioned) or 100 (for severely ill-conditioned problems), and accordingly  $d$  was set to be 1 or 2.

## 5.1 Surface Reconstruction

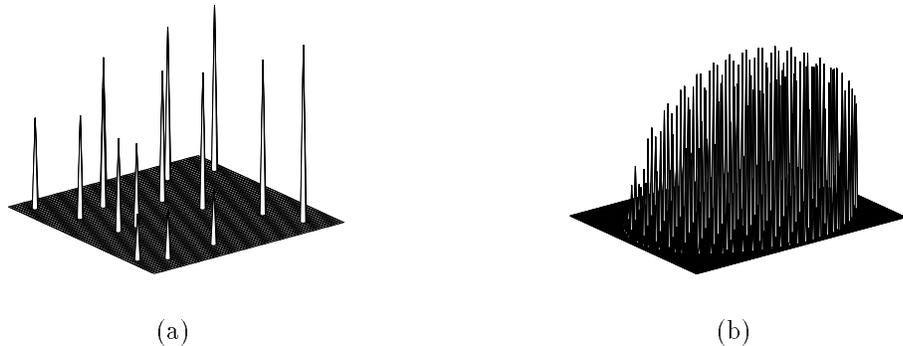


Figure 3: The sparse data sets used for the surface reconstruction, (a) synthetic data (b) real laser range data

We present results of several experiments with sparse data, and compare the convergence rate of our adaptive preconditioned conjugate gradient (APCG) algorithm with the hierarchical basis conjugate gradient (HCG) technique [22], the conjugate gradient (CG) algorithm [8], and the bi-orthogonal wavelet basis transfer scheme with Jaffard’s diagonal preconditioning [15]. In [32], the biorthogonal wavelet basis transfer is used in the preconditioning process. We found that this wavelet basis transfer *must be combined* with appropriate scaling to improve the convergence, which was not mentioned in [32]. In [15], Jaffard proposed the  $2^j$ , where  $j$  is the level, diagonal scaling in the wavelet basis as a preconditioning technique. Jaffard’s  $2^j$  diagonal scaling can be changed to  $2^{mj}$  scaling where  $m$  is a positive integer. We combined this variant of Jaffard’s diagonal scaling with the bi-orthogonal wavelet basis transfer and implemented it for conducting the convergence comparisons. We use all of

the aforementioned preconditioners in conjunction with a conjugate gradient algorithm and test their performance on one synthesized sparse data set (see figure 3) in the surface reconstruction experiments and one real range data set from a laser scanned ball. Subsequent experiments depict the convergence comparison of the aforementioned preconditioners applied to the SFS, and OF computation problems.

In the first experiment, we recover the underlying surface from the sparse data set in figure 3(a) using the controlled-continuity stabilizer with the parameters  $\rho$  and  $\tau$  being 1 and 0.1, respectively. The size of the problem is  $64 \times 64$ . Applying our adaptive preconditioner to this problem, we can see that the iterations converge to the true solution very fast. The computed solutions after 1, 5, 10 and 20 steps of conjugate gradient, HCG, and our APCG algorithms are illustrated in figure 4. With only five iterations of our APCG algorithm, the global shape of the surface is recovered. After ten iterations of our algorithm, the solution is very close to the true solution. Comparing these intermediate solutions in figure 4, we can see the solution after 20 iterations of HCG is still bumpy and the computed solution after 20 CG iterations is far from the true solution. In figure 5(a), we depict the convergence rate of the APCG algorithm in comparison to all the aforementioned iterative algorithms. Figure 5(b) depicts the experiments with a thin plate stabilizer. The convergence curves for our APCG algorithm, HCG, CG, and Yaou & Chang's bi-orthogonal basis transfer with Jaffard's diagonal preconditioning are shown. As can be seen, our APCG algorithm has the best convergence performance in both experiments.

We include the discontinuities in the above controlled-continuity stabilizer to test the performance of our APCG algorithm. The discontinuities are located along a pre-specified diagonal line. Figure 6 illustrates the convergence rates of our APCG, HCG( with 4 levels) and CG algorithms. The APCG algorithm still has the best convergence performance in this experiment. Since the inclusion of discontinuities makes the conditioning of the stiffness matrix worse and our preconditioner was not redesigned to account for the discontinuities in the stabilizer, the convergence rate in this experiment is slower than that of APCG without discontinuities. The computed solutions after 10, 20, 40 and 60 iterations of our APCG algorithm are depicted in figure 7.

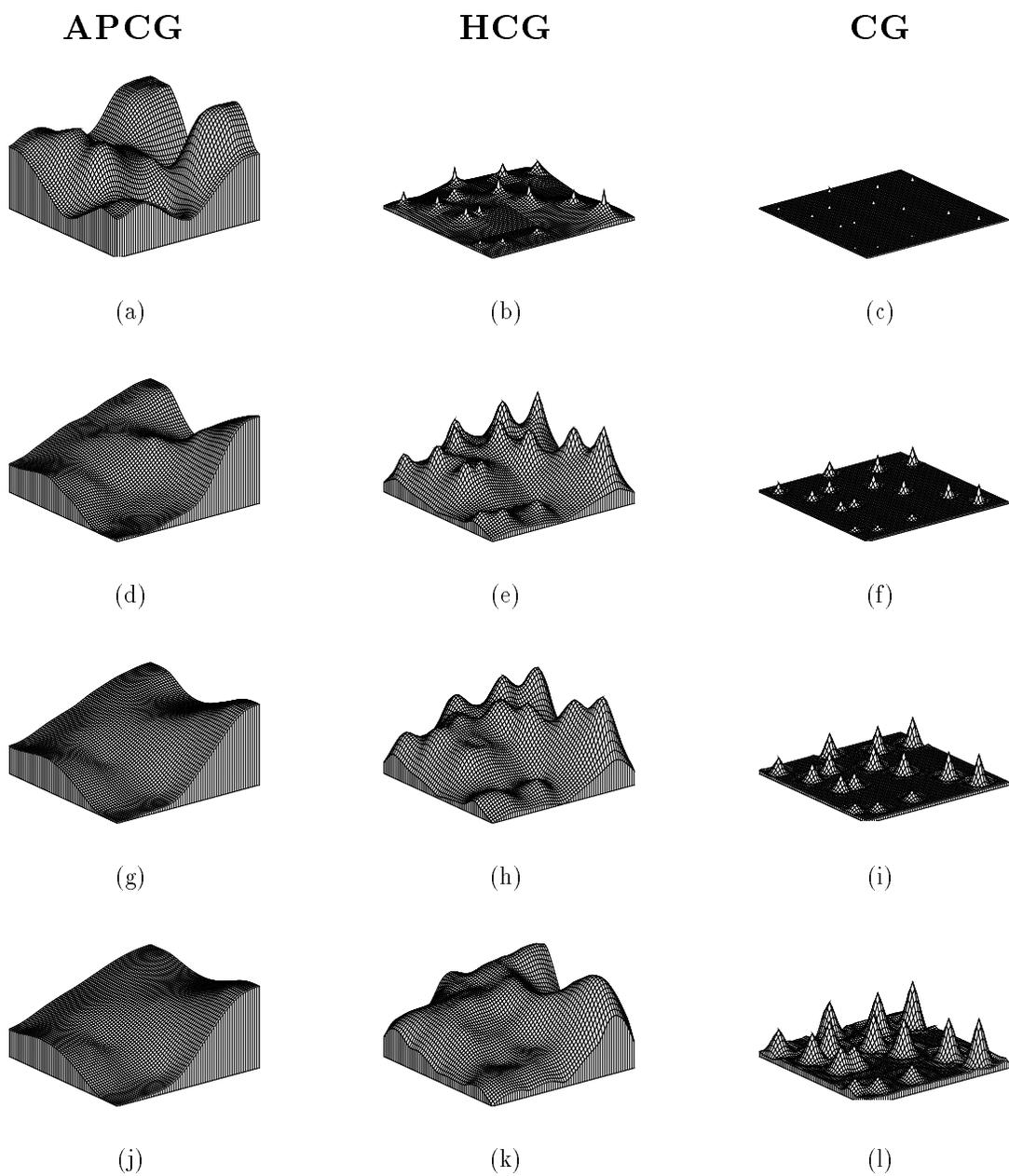
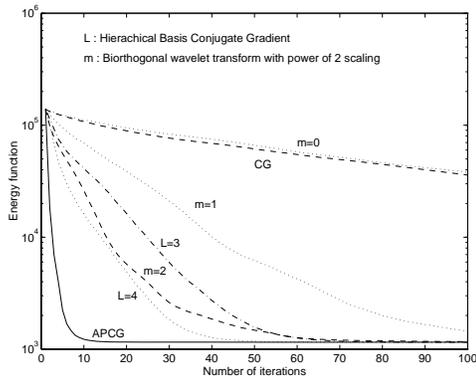
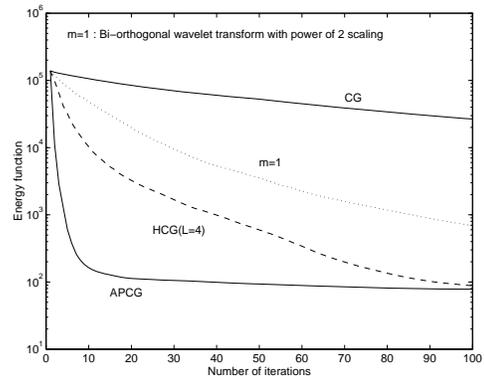


Figure 4: (a), (d), (g) & (j) are the computed solutions using the APCG algorithm, (b), (e), (h) & (k) are the computed solutions using HCG algorithm, (c), (f), (i) & (l) are the computed solutions using conjugate gradient algorithm ,after 1, 5, 10 and 20 iterations, respectively.



(a)



(b)

Figure 5: The convergence for different preconditioners in the preconditioned conjugate gradient algorithm in the surface reconstruction problem with the synthetic sparse data set and (a) with controlled-continuity stabilizer ( $\rho = 1, \tau = 0.1$ ) (b) with thin-plate stabilizer.

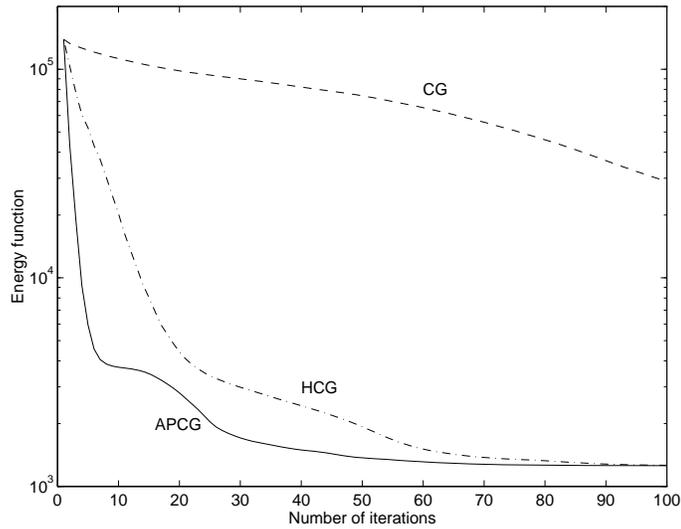


Figure 6: The convergence for different preconditioners in the preconditioned conjugate gradient algorithm for the surface reconstruction problem with discontinuities specified in the controlled-continuity stabilizer.

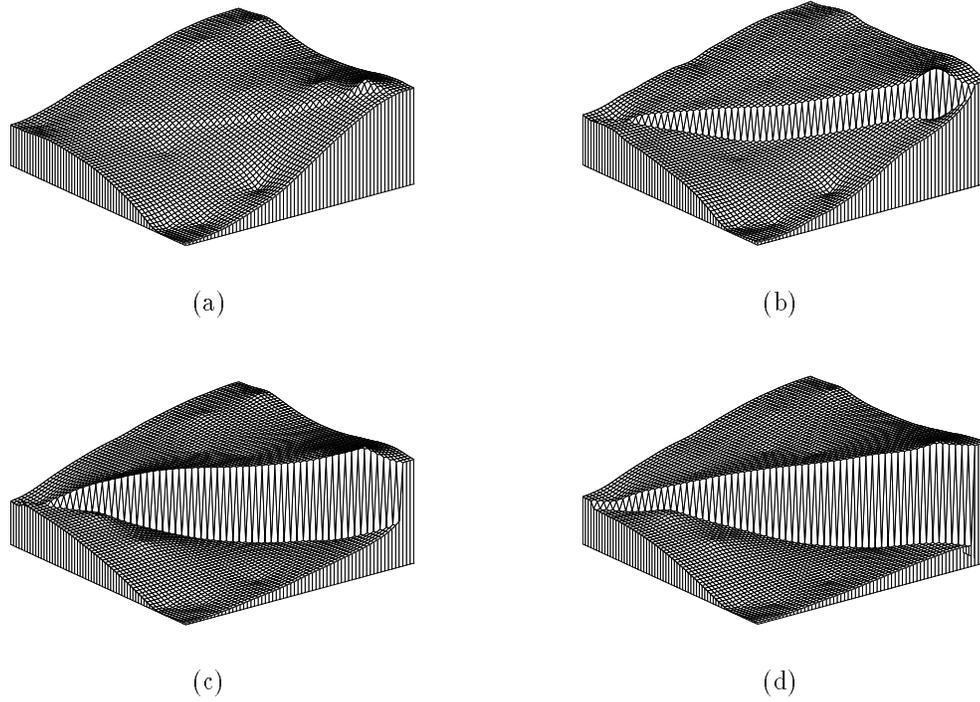


Figure 7: The computed solutions for the surface reconstruction with discontinuities using the APCG algorithm after (a) 10, (b) 20, (c) 40 and (d) 60 iterations, respectively.

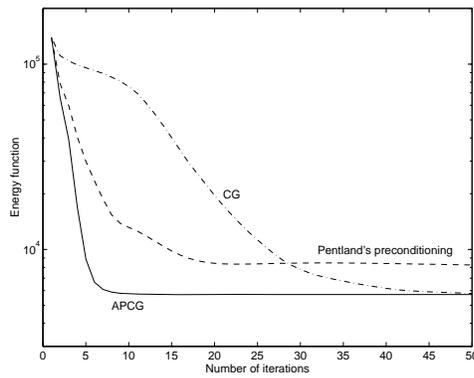


Figure 8: The convergence of Pentland's preconditioner in conjunction with the conjugate gradient algorithm is worse than that of conjugate gradient algorithm without preconditioning for the surface reconstruction problem with the synthetic sparse data set using a membrane stabilizer.

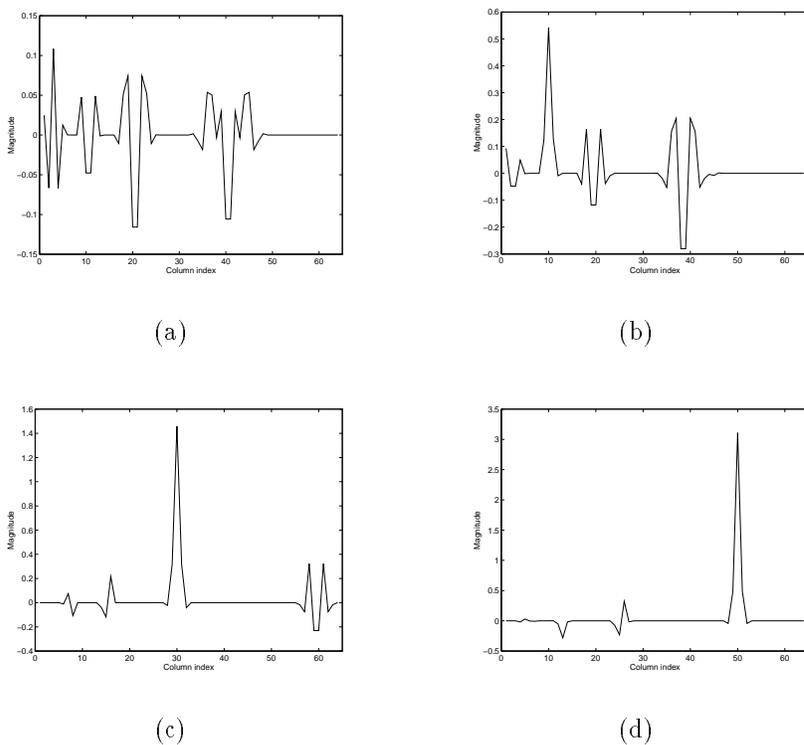


Figure 9: Row profile of the wavelet transformed membrane matrix  $\mathbf{K}_s$  of size  $64 \times 64$  along different rows, (a) row 3 (b) row 10 (c) row 30 (d) row 50. This figure establishes the falsity of the diagonal dominance behavior of  $\mathbf{K}_s$  claimed in [19].

We attempted to incorporate the preconditioner developed in Pentland’s [19] into the comparison however, we found that the convergence of conjugate gradient with this preconditioner is even slower than that of the CG algorithm (without preconditioning) for the very sparse data case. Note that the preconditioner was used in a gradient decent iteration in [19]. In our experiments, the preconditioners are used in conjunction with the conjugate gradient algorithm, which is known to be much more efficient than a gradient decent algorithm [8]. We use the sub-sampled data set in figure 3(a) with a membrane stabilizer in a  $32 \times 32$  problem for testing this preconditioner. The convergence curves for Pentland’s preconditioner with CG, CG and our APCG algorithms are shown in figure 8. We can see that Pentland’s preconditioning is worse than the conjugate gradient iterations without preconditioning. This is because the diagonal dominance behavior of  $\mathbf{K}_s$  after wavelet transform claimed in [19] doesn’t hold in general. Pentland’s preconditioner was constructed by approximating

the wavelet transform of the stiffness matrix by its diagonal entries based on the diagonal dominance assumption of  $\mathbf{K}_s$ . The falsity of the diagonal dominance claim is evident in figure 9 where we profile arbitrarily selected rows of the wavelet transformed membrane matrix  $\mathbf{K}_s$  of size  $(64 \times 64)$  for the 1D problem. In [19], the preconditioner was chosen as the diagonal of the stiffness matrix  $\mathbf{K}$  in the wavelet basis. Note that for the sparse data case, the matrix  $\mathbf{K}_s$  dominates  $\mathbf{K}_d$ . Therefore, bad approximations to the wavelet transform of  $\mathbf{K}_s$  will deteriorate the preconditioning effect and slow down the convergence speed.

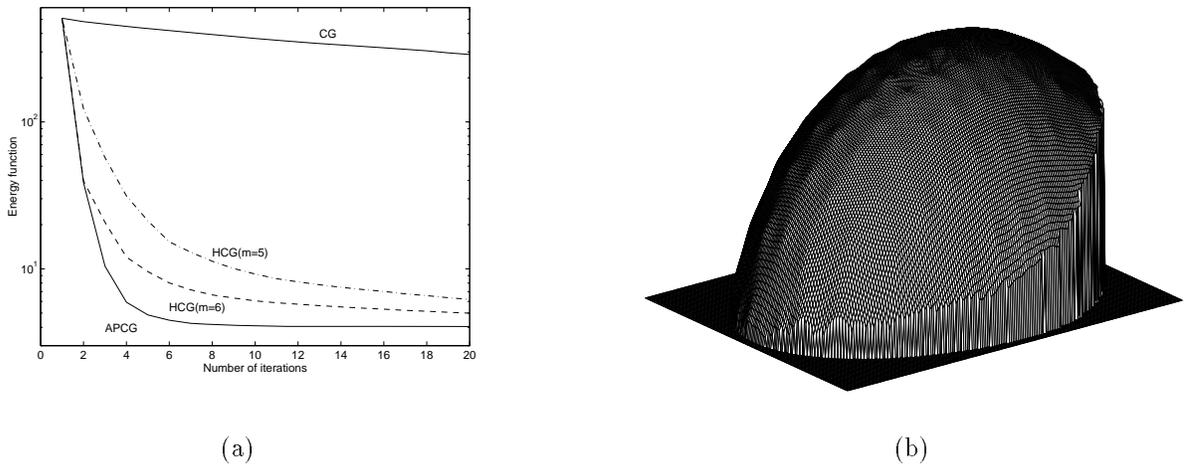


Figure 10: (a) The convergence for the APCG, HCG and CG algorithms for the surface reconstruction problem with the real laser range data set using a controlled-continuity stabilizer, (b) The computed solution after 10 APCG iterations.

The other data set in our surface reconstruction experiments is the real laser range data from a ball. Figure 3(b) shows the 148 randomly sampled data points from the original range data set generated by the MSU (Michigan State University) Pattern Recognition and Image Processing Lab's Technical Arts 100X scanner. The size of the problem domain is  $152 \times 193$ . The controlled-continuity stabilizer with  $\rho = 1$  and  $\tau = 0.1$  was used in this experiment. The discontinuities are specified along the boundary of the object. The discontinuities divide the problem domain into two non-overlapped regions, where the inside region corresponds to the object surface and the outside region corresponds to uniform background. Our adaptive preconditioning is used in the inside region only by enforcing

the solution values in the outside region to be zero after the preconditioning. The convergence curves for our APCG, HCG (with different levels), and CG algorithms are shown in figure 10(a). Our adaptive preconditioned CG algorithm still has the best convergence performance in this experiment with real laser range data. Very accurate solution is obtained after 10 iterations of the APCG algorithm, as shown in figure 10(b).

## 5.2 Shape from Shading

For the shape-from-shading experiment, we use the synthetic bump images with size  $64 \times 64$  used in [23, 11], as shown in figure 11. These two images are generated from the Lambertian reflectance model with the light source directions  $(p_s, q_s, 1)$  being  $(0.34, 0.3467, 1)$  and  $(-0.34, 0.3467, 1)$  for the left and right images, respectively. In this experiment, only the left image is taken as the input to the SFS algorithm to recover the shape, and the right image is used to monitor the closeness of the computed solution to the true solution.

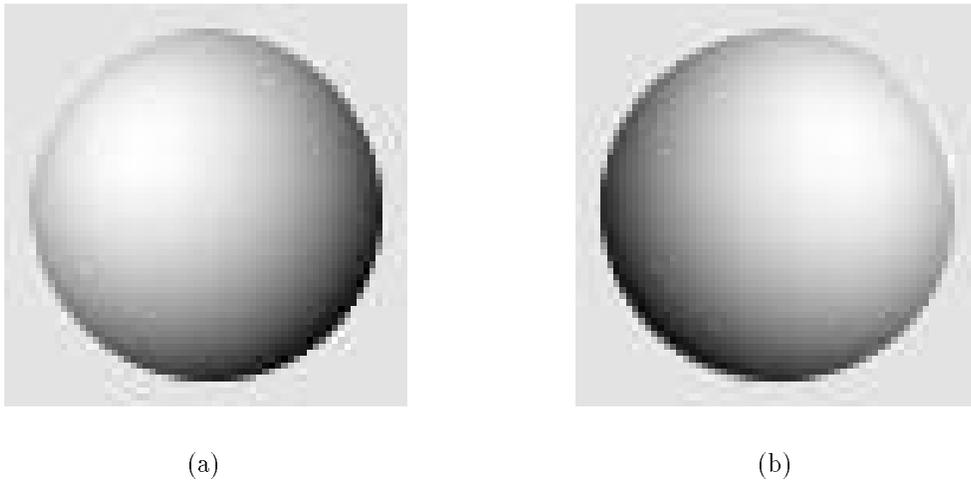
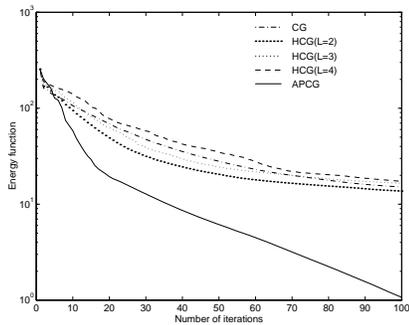
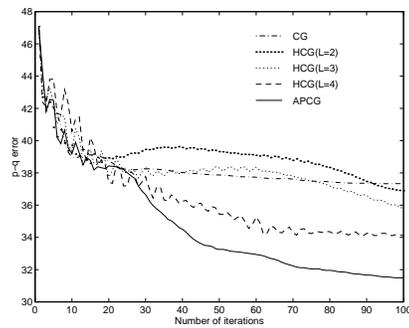


Figure 11: Shaded images of bump with different light source directions.

The construction of our preconditioner for the shape-from-shading problem is described in the previous section. The parameters  $\lambda$  and  $\mu$  are set to be 0.5 in this experiment. Without including

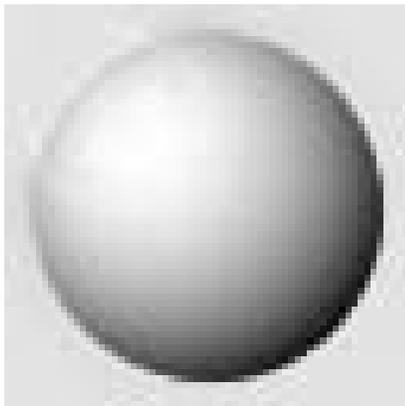


(a)

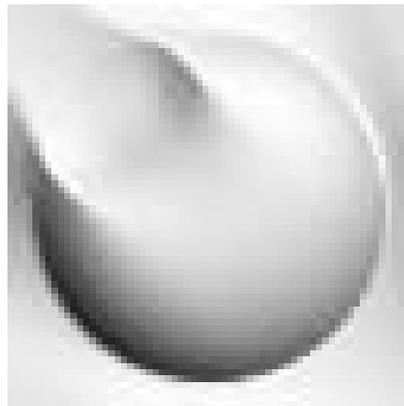


(b)

Figure 12: The convergence of the APCG, HCG, and CG algorithms for the shape-from-shading problem.



(a)

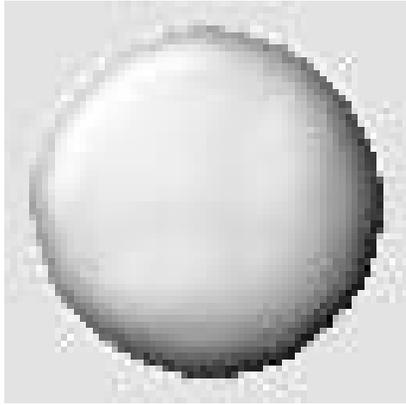


(b)

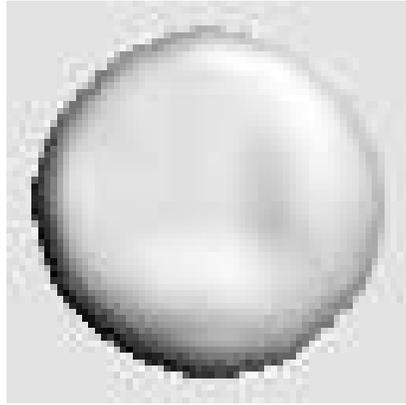
Figure 13: Shaded images of bump after 100 iterations of APCG algorithm without using any boundary condition.

any boundary condition in the energy function to be minimized, we compare the convergence of our APCG algorithm with the HCG and CG algorithms. The convergence curves for the energy function and the computed  $(p, q)$  error are shown in figure 12. Note that the  $(p, q)$  error is defined as  $\sum |p_i - p_i^*| + |q_i - q_i^*|$ , where  $(p_i, q_i)$  and  $(p_i^*, q_i^*)$  are the computed and true  $(p, q)$  at the  $i$ -th location, respectively. We can see the APCG algorithm has the best convergence behavior in figure 12(a) and (b). The computed solution after 100 iterations of the APCG algorithm is fed back to the reflectance model to generate the left and right bump images as shown in figure 13. Comparing these computed images to those in figure 11, we can see that this computed solution is far from the true solution in the upper left region, although its corresponding energy is close to 0. This is because the energy function for the SFS problem is nonconvex and gradient-based methods (e.g. CG-type methods) tend to get trapped in local minima. To obtain the true solution using the gradient-based method, we need to have an appropriate initial guess that is closer to the true solution. But it is difficult to find such initial guess that is guaranteed to converge to the true solution in the CG-type algorithms. In fact, we always use  $\mathbf{0}$  as the initial guess in our experiments.

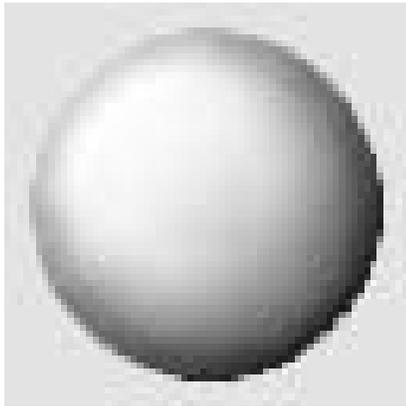
To resolve the above problem for minimizing the SFS energy function by gradient-based methods, we include the boundary conditions obtained from the occluding boundary [14] as an additional data constraint in the energy function to be minimized. By incorporating the boundary conditions into the SFS energy function, the CG-type algorithms can approach the true solution. In our experiment, we use the active contour model [16] to locate the occluding boundary at first. By using the property that the surface normal at a point on the occluding boundary is parallel to the normal of the silhouette in the image plane [14], we impose the Dirichlet boundary condition on  $(\mathbf{p}, \mathbf{q})$  along the occluding boundary. In addition, we use the occluding boundary to incorporate the discontinuities in  $(\mathbf{p}, \mathbf{q})$  or  $\mathbf{z}$  representing orientation discontinuities or depth discontinuities respectively. In this experiment, we incorporate the orientation discontinuities along the occluding contour. Figure 14 shows the left (north west) and right (north east) images generated from the computed solutions for our APCG algorithm after 10, 20 and 40 iterations. We can see that the computed solutions approach to the true solution after incorporating the boundary conditions. The computed gradient and height after



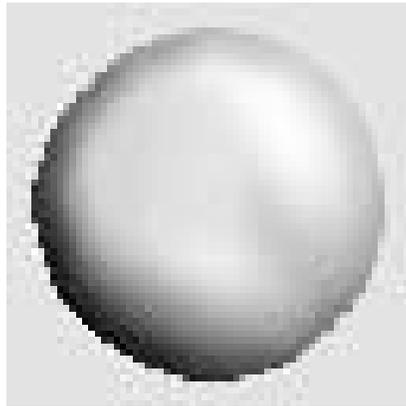
(a)



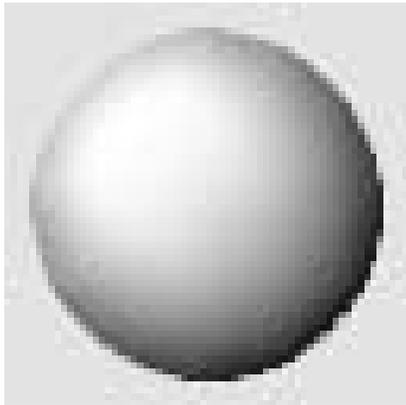
(b)



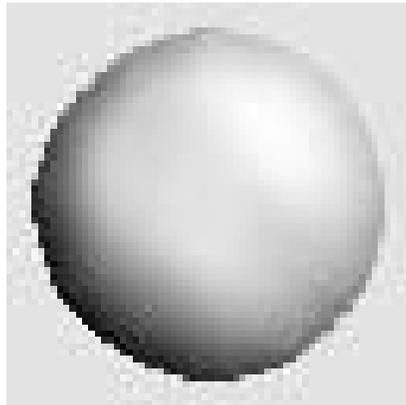
(c)



(d)



(e)



(f)

Figure 14: Shaded images of bump after (a) & (b) 10 iterations, (c) & (d) 20 iterations, (e) & (f) 40 iterations of the APCG algorithm with the occluding boundary condition.

40 iterations are shown in figure 15.

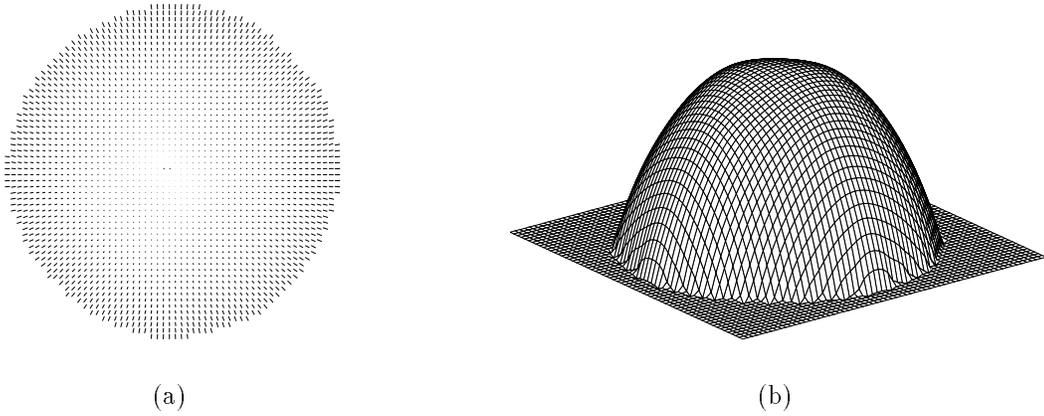


Figure 15: The computed (a) gradient and (b) height after 40 iterations of the APCG algorithm.

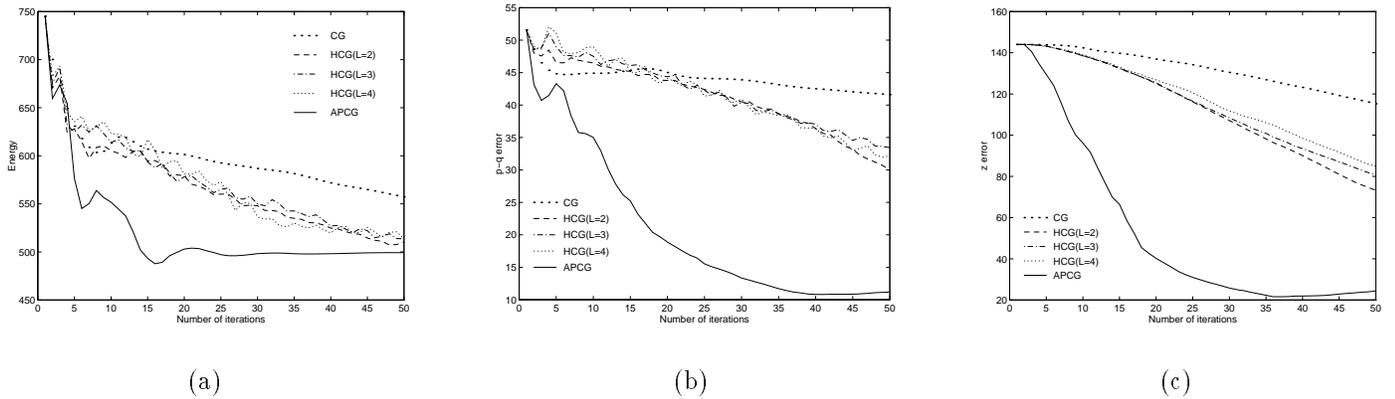


Figure 16: The convergence curves of the (a) energy, (b)  $\mathbf{p} - \mathbf{q}$  error, (c)  $\mathbf{z}$  error for the CG, HCG and APCG algorithm.

The convergence of our APCG algorithm was compared to the HCG and CG algorithms for the SFS problem with the Dirichlet boundary condition. The convergence curves for the energy,  $\mathbf{p} - \mathbf{q}$  error and  $\mathbf{z}$  error reduction are shown in figure 16. Our APCG outperforms the other methods in terms of convergence speed for all of the three different measures.

### 5.3 Optical Flow Computation

In section 2, we presented a new formulation of the optical flow computation problem. Our regularization-based formulation combines the region-based [13] and contour-based [10] gradient constraints. The gradient constraint for the brightness function  $E$  is active all over the image plane except in the neighborhood of the discontinuity locations, while the gradient constraint for the brightness function after the Laplacian of Gaussian operation, i.e.  $\nabla^2 G * E$ , is active at the discontinuity locations. In our implementation, we locate the zero-crossings of  $\nabla^2 G * E$  and label them as discontinuities [10]. Accurate estimation of  $\overline{E}_x$ ,  $\overline{E}_y$  and  $\overline{E}_t$  is necessary to obtain a reliable flow constraint. When  $\overline{E}$  is the brightness function, we use the Gaussian spatial-temporal pre-smoothing prior to applying the two-point central difference to approximate the partial derivatives. When  $\overline{E}$  is  $\nabla^2 G * E$ , a two-point central difference is applied to  $\overline{E}$  to approximate its partial derivatives. As discussed in section 2, a normalization of the gradient constraints is used to make the contribution of each (active) flow constraint uniform.

We present two experiments for the optical flow computation. The first experiment consists of a synthetic translating square image sequence; and the second is a **real image sequence** consisting of a rotating Rubic cube. One frame for each image sequence is shown in figure 17 . Each frame for the translating square and rotating Rubic cube image sequence is of size  $100 \times 100$  and  $256 \times 240$  respectively.

The construction of our adaptive preconditioner for the optical flow problem is described in section 4. This preconditioner is embedded in the conjugate gradient to accelerate the convergence speed for solving the linear system in the optical flow computation. We use the CG, HCG ( with various number of levels), and our APCG algorithms to solve the problem. Figure 19(a) and (b) shows the convergence rates of the three algorithms for the translating square and the rotating Rubic examples, respectively. We can see that our APCG algorithm has the best performance in convergence. Note that the HCG algorithm also converges quite fast because the linear system for the optical flow problem is not as ill-conditioned as the one in the surface reconstruction and shape-from-shading

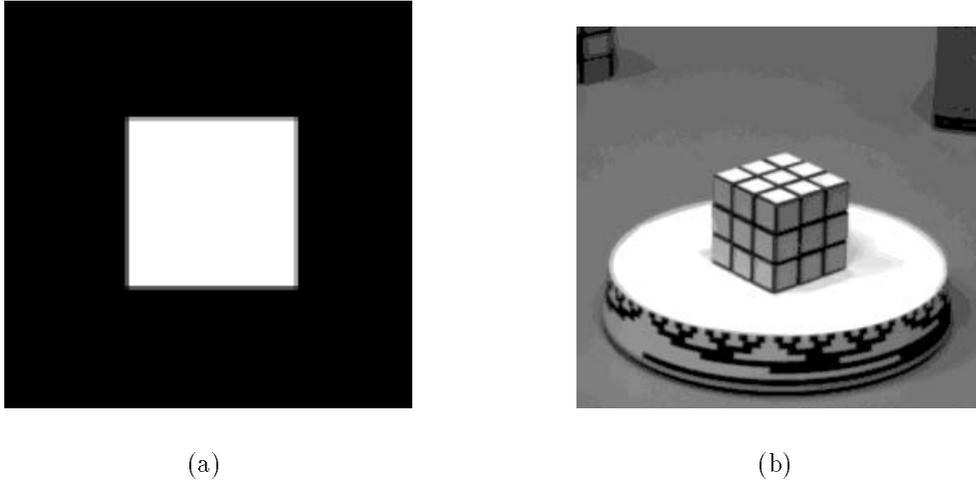


Figure 17: One frame from the (a) square image sequence and (b) rotating Rubic cube image sequence.

problems. Therefore, the difference between various preconditioning methods is not as pronounced for the optical flow problem as in the other two (SR & SFS) problems. Although the HCG algorithm with an appropriate number of levels  $L$ , exhibits fast convergence, it is difficult to determine the best  $L$  for a problem in advance. As for the APCG algorithm, the number of levels is always chosen to be the maximum possible which allows the linear convolution in the QMF implementation with the mirror reflection at the border to be carried out with ease.

In our examples, the regularization parameter  $\lambda$  was set to a value of 10 for the translating square example and 1 for the rotating Rubic cube experiment. For the former, the computed optical flows **after 10 iterations of the APCG algorithm** is shown in figure 18(a), and the **average angular error** is  $0.21^\circ$  with the standard deviation  $0.07^\circ$ . Compared to the results for the other gradient based methods reported in Barron et al. [1], our new formulation **gives the best solution** with 100% flow density. In fact, our adaptive preconditioner can also be used with the other gradient based regularization formulations of the optical flow problem. For the rotating Rubic cube experiment, the computed optical flow **after 15 iterations of our APCG algorithm** is shown in figure 18(b).

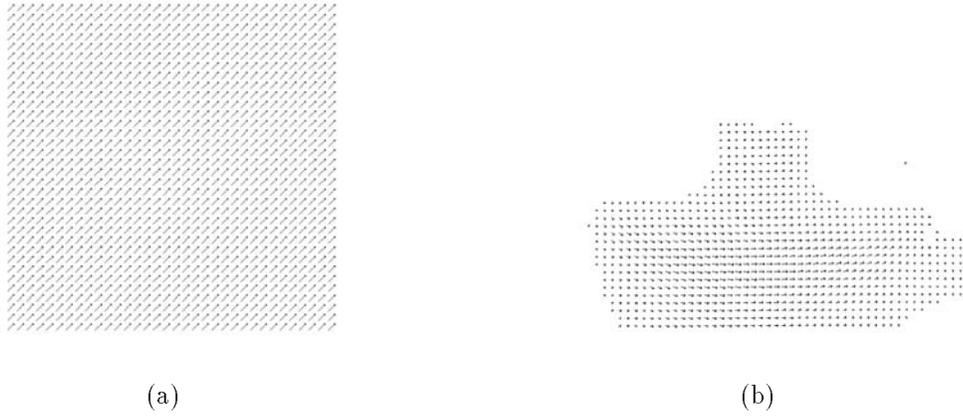


Figure 18: Computed flow of the (a) translating square and (b) rotating Rubic cube.

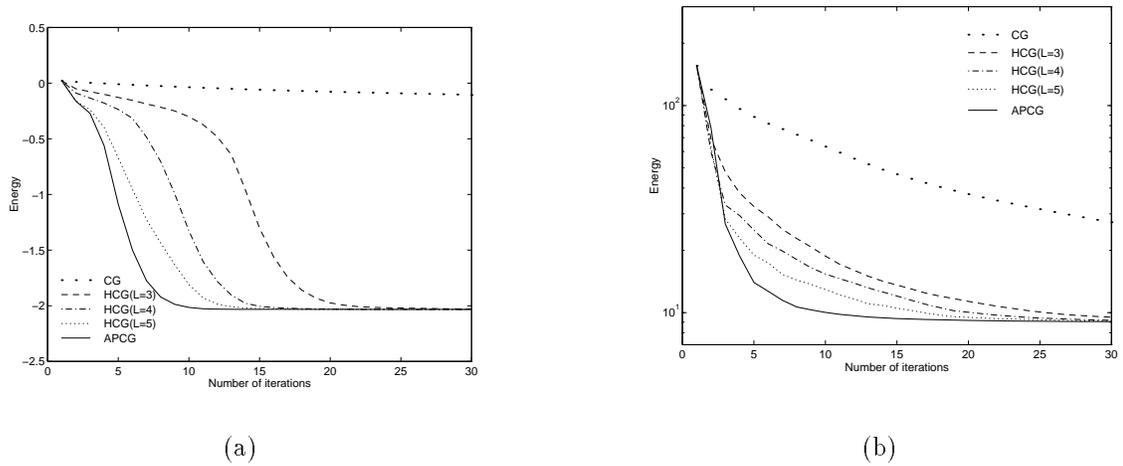


Figure 19: The convergence curves of the CG, HCG and APCG algorithms for (a) translating square (b) rotating Rubic image sequence.

## 6 Discussion and Conclusion

In this paper, we presented a novel physically-based adaptive preconditioning technique which was used in conjunction with a conjugate gradient algorithm to drastically improve the speed of convergence for solving various early vision problems. A preconditioner based on the membrane spline or the thin plate spline or a convex combination of the two was termed as a physically-based preconditioner for obvious reasons. The adaptation of the preconditioner to an early vision problem was achieved via the explicit use of the spectral characteristics of the regularization filter in conjunction with the data. This spectral function was used to modulate the frequency characteristics of a chosen wavelet basis leading to the construction of our preconditioner. The preconditioning technique was demonstrated for the surface reconstruction, shape from shading and optical flow computation problems. Through experiments, we demonstrated the superiority of our preconditioning method over existing methods.

Our adaptive preconditioning technique can be used in conjunction with any wavelet basis. The choice of the wavelet basis is crucial to achieve efficient preconditioning. The ideal wavelet needs to be well localized in spatial (or time) and frequency domain. Good spatial localization makes the QMF implementation very efficient, i.e., the wavelet transform can be accomplished very efficiently. Good localization in frequency domain of the wavelet is necessary to achieve fast convergence since the accuracy of spectral approximation to the regularization filter depends on the localization in frequency domain of the wavelet. But the localization in spatial and frequency domain cannot be arbitrarily small, because the product of their bandwidths is lower bounded by a constant  $\frac{1}{4\pi}$ , which is known as the uncertainty principle or Heisenberg inequality. This implies that there is a tradeoff between the spatial and frequency localization. Gaussian functions are optimal in the sense that they meet the bound with equality [7]. If we use the wavelet with the optimal property, the preconditioning should be more efficient.

In the construction of our adaptive preconditioner, we approximate the associated regularization filter by a linear shift-invariant filter in a wavelet basis, since the modulation matrix in our preconditioner gives uniform weighting to the wavelet basis functions corresponding to the same frequency

band. As discussed in section 3, the regularization filter is spatially varying in general, especially for the regularization problems with discontinuities or sparse data constraints. However, it is possible to construct a spatially varying filter in a wavelet basis to achieve a better approximation to the regularization filter. This will be the focus of our future research.

The adaptive preconditioner constructed in a wavelet basis can be applied to any regularization formulation in early vision. In fact, we can use similar construction of the adaptive preconditioner for the linear system discretized from any of the boundary value problems. The extension is straightforward by replacing the spectral characteristics of the membrane or thin-plate stabilizer by that of the associated differential operator. There is no restriction on the order of the partial differential equation. From the success of our adaptive preconditioner in the early vision problems, we believe the adaptive preconditioner will lead to drastic acceleration for the numerical solution of the boundary value problems.

## References

- [1] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. “Performance of optical flow techniques”. *Intl. Journ. of Comput. Vision*, **12**((1)):43–77, 1994.
- [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] B. L. Buzbee, F. W. Dorr, J. A. George, and G. H. Golub. “The direct solution of the discrete Poisson equation on irregular regions, ”. *SIAM Journal of Numerical Analysis*, 8(4):722–736, December 1971.

- [6] A.K. Chhabra and T.A. Grogan. “On Poisson solvers and semi- direct methods for computing area based optical flow”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(11), Nov. 1994.
- [7] D. Gabor. “Theory of communication”. *Journal of the IEE*, 93:429–457, 1946.
- [8] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 2nd edition, 1989.
- [9] W. E. L. Grimson. “an implementation of a computational theory of visual surface interpolation”. *Computer Vision, Graphics, Image Processing*, 22:39–69, 1983.
- [10] E. C. Hildreth. “Computations underlying the measurement of visual motion”. *Artificial Intelligence*, 23:309–354, 1984.
- [11] B. K. P. Horn. “Height from shading”. *Intl. Journal of Computer Vision*, **5**((**1**)):174–208, 1990.
- [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] K. Ikeuchi and B. K. P. Horn. “Numerical shape from shading and occluding boundaries”. *Artificial Intelligence*, 17:141–184, 1981.
- [15] S. Jaffard. “Wavelet methods for fast resolution of elliptic problems ”. *SIAM Journal of Numerical Analysis*, 29(4):965–986, August 1992.
- [16] M. Kass, A. Witkin, and D. Terzopoulos. “Snakes: active contour models”. *Intl. Journal of Computer Vision*, 1:321–331, 1987.

- [17] S. H. Lai and B. C. Vemuri. “An  $O(N)$  iterative solution to the Poisson equations in low-level vision problems,”. In *IEEE conference on Computer Vision & Pattern Recognition*, Seattle, Washington, June 1994.
- [18] S. G. Mallat. “A theory of multiresolution signal decomposition”. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-11(7):647–693, 1989.
- [19] A. Pentland. “Interpolation using wavelet basis”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-16(4):410–414, 1994.
- [20] O. Rioul and M. Vitterli. “Wavelets and signal processing”. *IEEE Signal Processing Magazine*, 8:14–38, 1991.
- [21] 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.
- [22] R. Szeliski. “fast surface interpolation using hierarchical basis functions”. *IEEE Trans. Pattern Anal. Machine Intell.*, 12(6):513–528, 1990.
- [23] R. Szeliski. “Fast shape from shading”. *CVGIP: Image Understanding*, 53(2):129–153, 1991.
- [24] D. Terzopoulos. “Image analysis using multigrid relaxation methods”. *IEEE Trans. Pattern Anal. Machine Intell.*, 8:129–139, 1986.
- [25] D. Terzopoulos. “Regularization of inverse visual problems involving discontinuities”. *IEEE Trans. Pattern Anal. Machine Intell.*, 8(4):413–424, 1986.
- [26] D. Terzopoulos. “The computation of visible-surface representations”. *IEEE Trans. Pattern Anal. Machine Intell.*, 10(4):417–438, 1988.
- [27] V. Torre and T. Poggio. “on edge detection”. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-8:147–163, 1986.

- [28] 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.
- [29] B. C. Vemuri and A. Radisavljevic. “Multiresolution stochastic hybrid shape models with fractal priors”. *ACM Trans. on Graphics*, 13(2):177–207, April 1994.
- [30] 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.
- [31] G. Whitten. “Scale space tracking of deformable sheet models for computational vision”. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-15(7):697–706, July 1993.
- [32] M. H. Yaou and W. T. Chang. “Fast surface interpolation using multiresolution wavelet transform”. *IEEE Trans. Pattern Anal. and Machine Intell.*, PAMI-16(7):673–688, 1994.
- [33] H. Yserentant. “On multi-level splitting of finite element spaces”. *Numerische Mathematik*, 49:379–412, 1986.