Surface Reconstruction via L^1-Minimization

Veselin Dobrev, Jean-Luc Guermond, and Bojan Popov

Department of Mathematics, Texas A&M University,
College Station, TX-77843, USA
{dobrev,guermond,popov}@math.tamu.edu

Abstract. A surface reconstruction technique based on the L^1-minimization of the variation of the gradient is introduced. This leads to a non-smooth convex programming problem. Well-posedness and convergence of the method is established and an interior point based algorithm is introduced. The L^1-surface reconstruction algorithm is illustrated on various test cases including natural and urban terrain data.

1 Introduction

In geometric modeling and image reconstruction, one often tries to *extract* a shape or recover a piece-wise smooth surface from a set of measurements. That is, one wants to find a surface that satisfies constraints or measurements and is visually good looking. The objectives could vary with the applications but the intuitive goal is to preserve the *shape* of the object. For example, one may want to reconstruct a convex body if the underlying data comes from a convex object, a flat surface if the data is locally flat, or preserve a particular structure of the level sets. Sometimes, this type of problems are solved by minimizing a L^p-norm of the curvature or the total variation of the gradient, see for example [1,2,3,4]. In this paper we take a different approach which we think is well suited for man made surfaces and Digital Elevation Maps (DEM). Namely, we minimize the total variation of the gradient of a function constructed on a finite element space satisfying interpolatory constraints. Similar minimization problems have been introduced by Lavery [5,1] and are hereafter referred to as the L^1-spline techniques. Minimizing the total variation of the gradient of a smooth function amounts to minimizing the L^1-norm of its second derivatives. The key observation from Lavery's work is that using the L^1-norm in the minimization process produces oscillation free surfaces.

In recent years, the idea of using the L^1-metric instead of the usual L^2 metric was exploited in many different areas with great success. For example, in compressed sensing [6,7] l^1-metric is used in the decoding step and in Partial Differential Equations the L^1-norm is used to measure the residual of the equation [8,9,10,11,12,13]. In all of the above applications, using L^1 is critical to obtain good numerical results and prove theoretical estimates.

One key ingredient in Lavery's work is the use of C^1-splines. The novelty of the approach in the present paper is to *relax* the C^1-smoothness on the finite element space which is used in the data reconstruction process. The discrete space is

S. Margenov, L.G. Vulkov, and J. Waśniewski (Eds.): NAA 2008, LNCS 5434, pp. 32–43, 2009.

composed of continuous finite elements with possibly discontinuous gradients. This is the natural discretization setting for functions that are in $W^{1,1}$ and whose gradient has bounded total variation.

The paper is organized as follows. In Section 2 we describe our scheme and in Section 3 we present different numerical tests for various types of data.

2 L^1-Minimization Problem

2.1 The Semi-discretized Functional

Let Ω be a bounded polygonal domain in $\mathbb{R}^2$ and let $\mathcal{T}_h$ be a partition of Ω composed of open triangles and quadrilaterals

$$\overline{\Omega} = \bigcup_{T \in \mathcal{T}_h} \overline{T}.$$

The mesh $\mathcal{T}_h$ is conforming in the sense that for any pair of distinct elements T, T' in $\mathcal{T}_h$, the intersection $T \cap T'$ is empty and $\overline{T} \cap \overline{T}'$ is either a common vertex or a common edge. For any element T in $\mathcal{T}_h$, we denote by h_T the diameter of T.

We introduce the discrete space X_h composed of continuous functions that are piecewise cubic on the mesh $\mathcal{T}_h$:

$$X_h = \{u \in \mathcal{C}(\overline{\Omega}) : u|_T \in \mathbb{P}_3 \text{ if } T \text{ is a triangle or,} \\ u|_T \in \mathbb{F}_T(\mathbb{Q}_3) \text{ if } T \text{ is a quadrilateral}, \forall T \in \mathcal{T}_h\} \quad (1)$$

where

$$\mathbb{P}_p = \left\{ \sum_{i=0}^{p} \sum_{j=0}^{p-i} c_{ij} x^i y^j : c_{ij} \in \mathbb{R} \right\}, \qquad \mathbb{Q}_{pq} = \left\{ \sum_{i=0}^{p} \sum_{j=0}^{q} c_{ij} x^i y^j : c_{ij} \in \mathbb{R} \right\}$$

and the mapping $\mathbb{F}_T$ is defined by

$$(\mathbb{F}_T \hat{q})(x) = \hat{q}(F_T^{-1}(x)), \quad \forall x \in T, \hat{q} \in \mathcal{C}([0,1]^2),$$

where F_T is the transformation that maps the reference unit square $(0,1)^2$ to the quadrilateral T. We henceforth denote $\mathbb{Q}_p := \mathbb{Q}_{pp}$.

The set of all the interior edges of the partition $\mathcal{T}_h$ is denoted by $\mathcal{F}_h^i$. Let $F \in \mathcal{F}_h^i$ be one of the interior edges and let $T, T' \in \mathcal{T}_h$ be the two elements whose intersection is $F = \overline{T} \cap \overline{T}'$. Also, let $\mathbf{n}_{TF}$ denote the normal vector to F pointing from T to T'. We define the jump of the normal derivative of a function u to be

$$[\![u_{\mathbf{n}}]\!]|_F = (\nabla u|_T) \cdot \mathbf{n}_{TF} + (\nabla u|_{T'}) \cdot \mathbf{n}_{T'F}.$$

The set of all the vertices of the triangulation $\mathcal{T}_h$ is be denoted by $\mathcal{V}_h$.

We now assume that we are given a real-valued function (data) taking values over the vertices of the mesh, $d_h : \mathcal{V}_h \to \mathbb{R}$. We denote by Y_h the affine set of functions in X_h interpolating the data

$$Y_h = \{u \in X_h : u(x) = d_h(x),\ \forall x \in \mathcal{V}_h\}\,.$$

Our goal is now to find a function in Y_h that oscillates as little as possible. We think of such a function as one that best fits the data map d_h. For this purpose, we introduce the following functional

$$\widetilde{J}_h(u) = \sum_{T\in\mathcal{T}_h} \int_T (|u_{xx}| + 2|u_{xy}| + |u_{yy}|) + \alpha \sum_{F\in\mathcal{F}_h^i} \int_F |[\![u_{\mathbf{n}}]\!]|\,, \quad u \in X_h$$

representing the total variation of the gradient of u with a weight, $\alpha > 0$. Note that $\widetilde{J}_h$ defines a semi-norm which vanishes if and only if its argument is a linear function on Ω.

The data reconstruction problem is formulated as follows: Find $u_h \in Y_h$ such that

$$\widetilde{J}_h(u_h) = \min_{v_h\in Y_h} \widetilde{J}_h(v_h). \tag{2}$$

Whenever we have at hand a family of meshes $(\mathcal{T}_h)_{h>0}$ and a corresponding family of data functions $(d_h)_{h>0}$, we say that a sequence $(v_h)_{h>0}$, with $v_h \in Y_h$, is a sequence of almost minimizers if there is a constant C_a, uniform with respect to h, so that

$$\widetilde{J}_h(u_h) \le C_a \min_{v\in Y_h} \widetilde{J}_h(v).$$

The following result clarifies the approximation properties of (2):

Proposition 1. *Assume that the mesh family $(\mathcal{T}_h)_{h>0}$ is shape regular. Assume that there is $u \in W^{2,1}(\Omega)$ so that $d_h(x) := u(x)$, $\forall x \in \mathcal{V}_h$. Let $(u_h)_{h>0}$ be a sequence of almost minimizers, then the following error estimates hold:*

$$\sum_{T\in\mathcal{T}_h} h_T^{j-2}|u - u_h|_{j,1,T} \le C|u|_{2,1,\Omega}, \qquad j = 0, 1.$$

2.2 Quadratures

The computation of the functional $\widetilde{J}_h$ is not practical due to the integration of absolute values. Therefore we discretize $\widetilde{J}_h$ by replacing the integrals with quadrature rules $I = \{(p, \omega)\}$ which we view as sets of pairs (p, ω) of points $p \in \mathbb{R}^2$ and weights $\omega > 0$. The terms of the functional $\widetilde{J}_h$ are be approximated using quadrature rules $I(S, \mathcal{L})$:

$$\int_S |\mathcal{L}u| \approx \sum_{(p,\omega)\in I(S,\mathcal{L})} \omega|(\mathcal{L}u)(p)|$$

where either $S \in \mathcal{T}_h$ and $\mathcal{L}$ is one of the linear operators $\{\partial_{xx}, 2\partial_{xy}, \partial_{yy}\}$, or $S \in \mathcal{F}_h^i$ and $\mathcal{L} = \alpha\,[\![\partial_{\mathbf{n}}]\!]$. We require that the integration rules $I(S, \mathcal{L})$ satisfy the following two conditions:

1. Be exact when the sign of the integrant $\mathcal{L}u$ does not change;
2. Give an approximation that is equivalent to the exact integral, i.e., there are constants c_1, c_2 independent of S, $\mathcal{L}$, and h so that:

$$c_1 \int_S |\mathcal{L}u| \leq \sum_{(p,\omega)\in I(S,\mathcal{L})} \omega |(\mathcal{L}u)(p)| \leq c_2 \int_S |\mathcal{L}u|, \qquad \forall u \in X_h.$$

In general the second condition requires the use of integration rules with more points than required by the first one. For example, if T is a triangle and $\mathcal{L} = \partial_{xx}$ then u_{xx} is linear and the midpoint rule satisfies the first condition but not the second. The following proposition gives a natural construction of quadrature rules satisfying both the above conditions under an easily verifiable assumptions:

Proposition 2. *Let $\widehat{S}$ be a (closed) reference element (e.g. triangle, square, segment), and T be an invertible* affine *transformation mapping $\widehat{S}$ to S. Also, let $\widehat{\mathcal{P}}$ be a finite-dimensional subspace of $\mathcal{C}(\widehat{S})$ (e.g. polynomials) and $\mathcal{P} = \mathbb{T}\,\widehat{\mathcal{P}}$ be its image under the transformation $\mathbb{T} : \mathcal{C}(\widehat{S}) \longrightarrow \mathcal{C}(S)$ defined by*

$$u(x) := \mathbb{T}(\hat{u})(x) = \hat{u}(T^{-1}(x)), \qquad \forall x \in S.$$

Let $\hat{I} = \{(\hat{p}_i, \hat{\omega}_i)\}_{i=1}^n$ be an integration rule with positive weights on $\widehat{S}$. If $\hat{I}$ is exact for every function in $\widehat{\mathcal{P}}$ and the quadrature points are such that

$$\left[\hat{u} \in \widehat{\mathcal{P}} \quad \text{and} \quad \hat{u}(\hat{p}_i) = 0, \quad i = 1, \ldots, n\right] \qquad \text{implies} \qquad \left[\hat{u}(\hat{x}) = 0, \quad \forall \hat{x} \in \widehat{S}\right],$$

then the integration rule $I = \{(p_i, \omega_i)\}_{i=1}^n$ with $p_i = T(\hat{p}_i)$ and $\omega_i = \frac{|S|}{|\widehat{S}|}\hat{\omega}_i$ (where $|\cdot|$ denotes the measure of the corresponding set) is exact for every function in $\mathcal{P}$ and

$$c_1 \int_S |u| \leq \sum_{i=1}^n \omega_i |u(p_i)| \leq c_2 \int_S |u|, \qquad \forall u \in \mathcal{P}$$

with constants $c_2 > c_1 > 0$ that depend on $\widehat{S}$ and $\widehat{P}$ but do not depend on the transformation T.

Based on the above proposition we use the following quadrature rules:

- When $S \in \mathcal{T}_h$ is a triangle and $\mathcal{L} \in \{\partial_{xx}, 2\partial_{xy}, \partial_{yy}\}$ then $\mathcal{L}(X_h|_S) = \mathbb{P}_1 = \widehat{\mathcal{P}} = \mathcal{P}$ and therefore the 3-point quadrature rule using the midpoints of the sides of the triangle satisfies the conditions of the proposition (this rule is exact for $\mathbb{P}_2$).
- When $S \in \mathcal{T}_h$ is a rectangle with sides parallel to the coordinate axes we use three different quadrature rules for the three different second derivatives. For $\mathcal{L} = \partial_{xx}$ we have $\mathcal{L}(X_h|_S) = \mathbb{Q}_{1,3} = \widehat{\mathcal{P}} = \mathcal{P}$ and therefore we could use the 2×4 tensor product Gaussian rule; however, numerical experiments show some undesired oscillations which can be avoided by using the 3×4 tensor product Gaussian rule. For $\mathcal{L} = 2\partial_{xy}$ we have $\mathcal{L}(X_h|_S) = \mathbb{Q}_{2,2} = \widehat{\mathcal{P}} = \mathcal{P}$ and we use the 3×3 tensor product Gaussian rule. For $\mathcal{L} = \partial_{yy}$, $\mathcal{L}(X_h|_S) = \mathbb{Q}_{3,1} = \widehat{\mathcal{P}} = \mathcal{P}$ and we use the 4×3 tensor product Gaussian rule.

- When $S \in \mathcal{T}_h$ is not a rectangle with sides parallel to the coordinate axes we have $\widehat{\mathcal{P}} \neq \mathcal{P}$ and it is more convenient to replace the second derivatives in $\widetilde{J}_h$ by second derivatives in directions parallel to the sides of S. This case is not considered in the numerical experiments reported in this paper.
- When $S \in \mathcal{F}_h^i$ and $\mathcal{L} = \alpha [\![\partial_{\mathbf{n}}]\!]$ we have two cases: 1) S is the edge of two triangles and 2) S is a side in a quadrilateral. $\mathcal{P}$ and $\widehat{\mathcal{P}}$ are composed of one-dimensional quadratic polynomials in the first case and cubic polynomials in the second case. Therefore, we use the 3 point Gaussian rule in the first case and the 4 point Gaussian rule in the second.

Using the above quadrature rules we obtain the approximate functional

$$J_h(u) = \sum_{\substack{T\in\mathcal{T}_h \\ \mathcal{L}\in\{\partial_{xx},2\partial_{xy},\partial_{yy}\}}} \sum_{(p,\omega)\in I(T,\mathcal{L})} \omega|(\mathcal{L}u)(p)| + \alpha \sum_{F\in\mathcal{F}_h^i} \sum_{(p,\omega)\in I(F,[\![\partial_{\mathbf{n}}]\!])} \omega|\,[\![u_{\mathbf{n}}]\!](p)|.$$

Note that J_h defines a semi-norm on X_h which is equivalent to that induced by $\widetilde{J}_h$ with constants independent of h.

The fully discretized version of problem (2) is the following: Find $u_h \in Y_h$ such that

$$J_h(u_h) = \min_{v_h\in Y_h} J_h(v_h). \tag{3}$$

2.3 Matrix Formulation

Let $\{\phi_i\}_{i=1}^{\hat{n}}$ be a basis for X_h. The functional J_h can be re-written as follows

$$J_h(u) = |\widehat{A}x|_1 \qquad \text{where} \qquad x \in \mathbb{R}^{\hat{n}} : u = \sum_{i=1}^{\hat{n}} x_i\phi_i$$

and the entries of the matrix $\widehat{A}$ are given by

$$\widehat{A}_{ij} = \omega_i(\mathcal{L}_i\phi_j)(p_i) \qquad i = 1,\dots,m \quad j = 1,\dots,\hat{n}.$$

Here $\{(p_i,\omega_i)\}_{i=1}^m$ is an enumeration of all the quadrature points (and weights) in all the quadrature rules used in the discretization of $\widetilde{J}_h$ and $\mathcal{L}_i$ is the linear operator corresponding to the quadrature rule. Thus, the total number of quadrature points is given by

$$m = \sum_{\substack{T\in\mathcal{T}_h \\ \mathcal{L}\in\{\partial_{xx},2\partial_{xy},\partial_{yy}\}}} \#\left(I(T,\mathcal{L})\right) + \sum_{F\in\mathcal{F}_h^i} \#\left(I(F,[\![\partial_{\mathbf{n}}]\!])\right)$$

where $\#(I)$ denotes the cardinal number of I.

Let us further assume that $\{\phi_i\}$ is the standard nodal basis for X_h and the basis functions corresponding to the vertices in $\mathcal{V}_h$ are the first $\hat{n}_1$ functions $\phi_1,\dots,\phi_{\hat{n}_1}$. The matrix $\widehat{A}$ can be written in 1×2 block form $\widehat{A} = (\widehat{A}_1 A)$ where $\widehat{A}_1$ is $m\times\hat{n}_1$ and A is $m\times n$, $(n = \hat{n}-\hat{n}_1)$. Let $d \in \mathbb{R}^{\hat{n}_1}$ be the vector representing

the data d_h at the vertices and set $b = -\widehat{A}_1 d$. Then the discrete problem (3) can be re-written as follows: Find $x \in \mathbb{R}^n$ such that

$$|Ax - b|_1 = \min_{y \in \mathbb{R}^n} |Ay - b|_1. \tag{4}$$

It can be shown that A is full rank.

2.4 Discrete Problem

In this section we study properties of ℓ_1-minimization problems of generic form (4). Let A be an $m \times n$ real matrix ($m > n$) and $b \in \mathbb{R}^m$. We define the Lagrangian

$$L(x, \lambda) = (b - Ax)^t \lambda, \qquad x \in \mathbb{R}^n, \lambda \in \mathbb{R}^m$$

and the primal and dual functions, f and g, respectively

$$f(x) = \max_{\substack{\lambda \in \mathbb{R}^m \\ |\lambda|_\infty \le 1}} L(x, \lambda) = |b - Ax|_1$$

$$g(\lambda) = \min_{x \in \mathbb{R}^n} L(x, \lambda) = \begin{cases} b^t \lambda & A^t \lambda = 0 \\ -\infty & A^t \lambda \neq 0. \end{cases}$$

It is clear that for all $x \in \mathbb{R}^n$ and all $\lambda \in \mathbb{R}^m$, $|\lambda|_\infty \le 1$ we have

$$f(x) \ge L(x, \lambda) \ge g(\lambda).$$

The primal problem is defined to be

$$\text{minimize} \quad f(x) = |b - Ax|_1 \tag{5}$$

and the dual problem is defined to be

$$\begin{aligned} &\text{maximize} \quad g(\lambda) = b^t \lambda \\ &\text{subject to} \quad A^t \lambda = 0, |\lambda|_\infty \le 1. \end{aligned} \tag{6}$$

Proposition 3. *(Strong duality) For any pair of solutions x^* and λ^* to (5) and (6), respectively, we have $f(x^*) = g(\lambda^*)$.*

Corollary 1. *If x^* is a solution of (5) and $(b - Ax^*)_i \neq 0$ for some index i then every solution λ^* of (6) satisfies $\lambda_i^* = \operatorname{sign}(b - Ax^*)_i$. In particular, if λ^* is a solution of (6) and $|\lambda_i^*| < 1$ then for every solution x^* of (5) we have $(b - Ax^*)_i = 0$.*

We now assume that A and b have the following block structure

$$A = \begin{pmatrix} A_1 \\ \alpha A_2 \end{pmatrix} \qquad b = \begin{pmatrix} b_1 \\ \alpha b_2 \end{pmatrix}$$

which is exactly the structure they have in problem (4) where A_2 and b_2 correspond to the rows generated by the terms $\int_F |[\![u_{\mathbf{n}}]\!]|$, $F \in \mathcal{F}_h^i$. The primal function has the form

$$f(x) = |b - Ax|_1 = |b_1 - A_1 x|_1 + \alpha |b_2 - A_2 x|_1.$$

Proposition 4. *Assume the rows of A_2 are linearly independent. Then there exists a number $\bar{\alpha}$ such that when $\alpha > \bar{\alpha}$ every solution x^* of (5) satisfies*

$$b_2 - A_2 x^* = 0.$$

Proof. We will show that when α is large enough the feasible set of the dual problem (6) (and therefore any solution) satisfies $|\lambda_2|_\infty < 1$ which, in view of Corollary 1, implies the proposition. Indeed, if λ is dual feasible we have

$$0 = A^t \lambda = A_1^t \lambda_1 + \alpha A_2^t \lambda_2.$$

The assumption on A_2 implies the existence of right inverse R of A_2:

$$A_2 R = I \qquad \text{or} \qquad R^t A_2^t = I$$

and thus we have $\lambda_2 = -\frac{1}{\alpha} R^t A_1^t \lambda_1$. Now, if we define $\bar{\alpha} = |R^t A_1^t|_\infty$ and take $\alpha > \bar{\alpha}$ we get

$$|\lambda_2|_\infty = \frac{1}{\alpha} |R^t A_1^t \lambda_1|_\infty \leq \frac{1}{\alpha} |R^t A_1^t|_\infty |\lambda_1|_\infty < 1.$$

Proposition 5. *There exists a number $\bar{\alpha}$ such that when $\alpha > \bar{\alpha}$ and $b_2 \in \mathrm{Im} A_2$ every solution x^* of (5) satisfies $b_2 - A_2 x^* = 0$.*

Proof. Let $\widetilde{A}_2$ denote the matrix whose rows are a maximal linearly independent set of rows of A_2. Without loss of generality we can write

$$A_2 = \begin{pmatrix} \widetilde{A}_2 \\ A_3 \end{pmatrix} \qquad b_2 = \begin{pmatrix} \tilde{b}_2 \\ b_3 \end{pmatrix}.$$

We have the following property: if $b_2 \in \mathrm{Im} A_2$ and $\widetilde{A}_2 x = \tilde{b}_2$ then $A_3 x = b_3$. Let us now define

$$\widetilde{A} = \begin{pmatrix} A_1 \\ \alpha \widetilde{A}_2 \end{pmatrix} \qquad \tilde{b} = \begin{pmatrix} b_1 \\ \alpha \tilde{b}_2 \end{pmatrix}$$

and consider the reduced minimization problem

$$\text{minimize} \quad \tilde{f}(x) = |\tilde{b} - \widetilde{A} x|_1 = |b_1 - A_1 x|_1 + \alpha |\tilde{b}_2 - \widetilde{A}_2 x|_1 \tag{7}$$

obtained from (5) by replacing A and b with $\widetilde{A}$ and $\tilde{b}$, respectively. Since the rows of $\widetilde{A}_2$ are linearly independent we can apply the previous proposition to this problem and conclude that for $\alpha > \bar{\alpha}$ every solution $\tilde{x}$ of (7) satisfies $\widetilde{A}_2 \tilde{x} = \tilde{b}_2$. We now assume that $\alpha > \bar{\alpha}$ and $b_2 \in \mathrm{Im} A_2$, and we want to show that problems (5) and (7) are equivalent. First we note that $\forall x \in \mathbb{R}^n$

$$\begin{aligned} \tilde{f}(x) &= |b_1 - A_1 x|_1 + \alpha |\tilde{b}_2 - \widetilde{A}_2 x|_1 \\ &\leq |b_1 - A_1 x|_1 + \alpha |\tilde{b}_2 - \widetilde{A}_2 x|_1 + \alpha |b_3 - A_3 x|_1 = f(x) \end{aligned}$$

and therefore for any two solutions x^* and $\tilde{x}$ of (5) and (7), respectively, we have $\tilde{f}(\tilde{x}) \leq f(x^*)$. Since $\widetilde{A}_2\tilde{x} = \tilde{b}_2$ and we assumed that $b_2 \in \mathrm{Im}A_2$ we conclude that $A_3\tilde{x} = b_3$ and therefore

$$f(\tilde{x}) = \tilde{f}(\tilde{x}) \leq f(x^*) \leq f(\tilde{x})$$

which shows that $\tilde{f}(\tilde{x}) = f(x^*)$ and $\tilde{x}$ is a solution to (5). For $\tilde{f}(x^*)$ we have

$$\tilde{f}(x^*) \leq f(x^*) = \tilde{f}(\tilde{x}) \leq \tilde{f}(x^*)$$

which shows that $\tilde{f}(x^*) = \tilde{f}(\tilde{x})$ and therefore x^* is a solution to (7). Since we already saw that $\widetilde{A}_2\tilde{x} = \tilde{b}_2$ and $A_3\tilde{x} = b_3$, that is $A_2\tilde{x} = b_2$, and since $\tilde{x}$ was an arbitrary solution to (7) (or as we just proved, to (5)) this completes the proof.

Corollary 2. *Assume that all elements of the mesh $\mathcal{T}_h$ are quadrilaterals. Then there exists $\bar{\alpha}$ such that when $\alpha > \bar{\alpha}$ every solution u_h to (3) is in $\mathcal{C}^1(\overline{\Omega})$.*

Proof. Note that $u_h \in \mathcal{C}^1(\overline{\Omega})$ is equivalent to $A_2x = b_2$ where (d, x) is the coefficient vector of u_h. Thus, $b_2 \in \mathrm{Im}A_2$ is equivalent to the existence of $v_h \in Y_h \cap \mathcal{C}^1(\overline{\Omega})$. In the case of quadrilateral elements such v_h can be constructed using Bogner-Fox-Schmit type interpolation in which one can prescribe not only the values of the function at the vertices but also its gradient and mixed second derivative. For triangular meshes, all solutions u_h are $\mathcal{C}^1$ if and only if the data and the mesh allow it, that is $Y_h \cap \mathcal{C}^1(\overline{\Omega}) \neq \emptyset$.

Remark 1. *In the above proof the value of $\bar{\alpha}$ is not a priori uniform with respect to the typical mesh-size h. However, numerical tests indicate that using $\alpha = 5$ guarantees $\mathcal{C}^1$-smoothness independently of h.*

2.5 Primal-Dual Interior-Point Method

We now describe an approach for solving the minimization problem (4). First, we reformulate (4) as a linear programming problem: Find $y \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$ so that

$$\text{minimize } y^t 1 = \sum_{i=1}^{m} y_i, \qquad \text{subject to } \begin{cases} y \geq b - Ax \\ y \geq Ax - b. \end{cases}$$

Then we apply the primal-dual interior-point method described in [14,15].

After some simplifications, which we omit here, the above problem is solved using the following algorithm:

input: A, b, x, λ; μ, ϵ
$r = b - Ax$;
$a = (|r|_1 - r^t\lambda)/m$; $\quad y_i = |r_i| + a$, $i = 1, \ldots, m$;
while $(|r|_1/(r^t\lambda) - 1 \geq \epsilon)$
$\quad t^{-1} = (y^t 1 - r^t\lambda)/(2m\mu)$;
$\quad s_1 = y + r$; $\quad s_2 = y - r$;
$\quad d_1 = (1 - \lambda)/(2s_1)$; $\quad d_2 = (1 + \lambda)/(2s_2)$;

$d = 4d_1d_2/(d_1 + d_2)$;
$v = t^{-1}(s_2^{-1} - s_1^{-1}) + (d_2 - d_1)/(d_1 + d_2)[1 - t^{-1}(s_1^{-1} + s_2^{-1})]$;
$w = A^t v$;
$\Delta x = (A^t \mathrm{diag}(d) A)^{-1} w$;
$v = A\Delta x$;
$\Delta y = [-1 + t^{-1}(s_1^{-1} + s_2^{-1}) + (d_1 - d_2)v]/(d_1 + d_2)$;
$\Delta\lambda = -\lambda + t^{-1}(s_2^{-1} - s_1^{-1}) - (d_1 + d_2)v + (d_1 - d_2)\Delta y$;
$s = \max\{\sigma \in (0, 2] : \lambda + \sigma\Delta\lambda \geq -1, \lambda + \sigma\Delta\lambda \leq 1,$
$\qquad y + \sigma\Delta y \geq r - \sigma v, y + \sigma\Delta y \geq -r + \sigma v\}$;
$s = \min\{1, 0.99s\}$;
$x = x + s\Delta x; \quad y = y + s\Delta y; \quad r = r - sv; \quad \lambda = \lambda + s\Delta\lambda$;
end while
output: x, λ;

The input parameter μ is a positive real number (we use $\mu = 10$) and ϵ is a given tolerance. The initial input value of the dual variable λ is assumed to be strictly dual feasible, that is $A^t\lambda = 0$ and $|\lambda|_\infty < 1$ (we use $\lambda = 0$). In the algorithm, a, t, and s are scalar variables; $r, y, d, v, \Delta y, \Delta\lambda \in \mathbb{R}^m$; $w, \Delta x \in \mathbb{R}^n$; the vectors s_1, s_2, d_1, d_2 do not need to be stored since their components can be evaluated one by one when needed (one time when computing d and v, and another time when computing Δy and $\Delta\lambda$). All operations in the definitions of $d_1, d_2, d, v, \Delta y$, and $\Delta\lambda$ are component-wise. We use $\mathrm{diag}(d)$ to denote the diagonal matrix with main diagonal given by the vector d.

It can be shown that all vectors λ generated by this algorithm are strictly dual feasible provided that the input λ is strictly dual feasible. Thus, the stopping criterion we use guarantees that

$$f(x) - f(x^*) \leq f(x) - g(\lambda) = |r|_1 - r^t\lambda < \epsilon\, r^t\lambda \leq \epsilon f(x^*)$$

which means that x is an almost minimizer for (4) with a tolerance $1 + \epsilon$.

The most expensive step at each iteration of the **while** loop is the solution of the equation for Δx. Since direct solution methods are not practical for large n, we use an iterative method to solve the linear system approximately. In the resulting algorithm the vectors λ do not satisfy $A^t\lambda = 0$. However, numerically we observe that solving iteratively with relative tolerance $\epsilon/10$ produces results that are very similar the results obtained by solving almost exactly. The iterative method we use is the preconditioned conjugate gradient (PCG) method with a simple symmetric Gauss-Seidel preconditioner.

3 Numerical Examples

We illustrate our data reconstruction technique in this section. In all numerical experiments Ω is the unit square and we use a uniform rectangular mesh with equal step size in both x and y directions. The tolerance in the interior-point (IP) method is $\epsilon = 10^{-2}$ and the linear systems for Δx are solved with relative

tolerance 10^{-3}. The initial approximation for the vector x in the IP method is obtained from the $\mathbb{Q}_1$ interpolant of the data.

3.1 Test 1: Piece-Wise Smooth Data

The data for this set of experiments is obtained from a single function

$$u(x, y) = f(\max\{|x - 1/2|, |y - 1/2|\}),$$

where

$$f(r) = \begin{cases} 5/3 & r \in [0, 1/8] \\ 1 & r \in (1/8, 5/16] \\ 16(1/2 - r)/3 & r \in (5/16, 1/2] \end{cases}$$

and we use meshes with step size, h, varying from 1/16 to 1/256. Note that $u(x, y)$ is discontinuous at $\Gamma_1 = \{r = 1/8\}$ and its gradient also has jumps at $\Gamma_2 = \{r = 5/16\}$ and at $\Gamma_3 = (\{x = y\} \cup \{x + y = 1\}) \cap \{5/16 \leq r \leq 1/2\}$. Away from those discontinuities the function is linear. Figure 1 shows the two reconstructed surfaces obtained with $\alpha = 3$ and $\alpha = 5$ on 16×16 mesh. The solution obtained with $\alpha = 5$ is $\mathcal{C}^1$ everywhere and that obtained with $\alpha = 3$ is $\mathcal{C}^1$ almost everywhere but around the edges defined by Γ_3. In Table 1 (left side), we present results for the convergence of the IP method as we refine the mesh. We see a very small increase in the number of IP iterations of order $\ln(1/h)$. The total number of PCG iterations is given along with the increase in those numbers from one level to the next and we multiply that ratio by 4 which roughly gives the increase in the computational cost per level. If we compare these ratios with the actual increase in computing time, we see that both are fairly close. These numbers indicate an order $\mathcal{O}(n^\beta)$, $\beta = \ln(6)/\ln(4) \approx 1.29$, for the computational complexity and time.

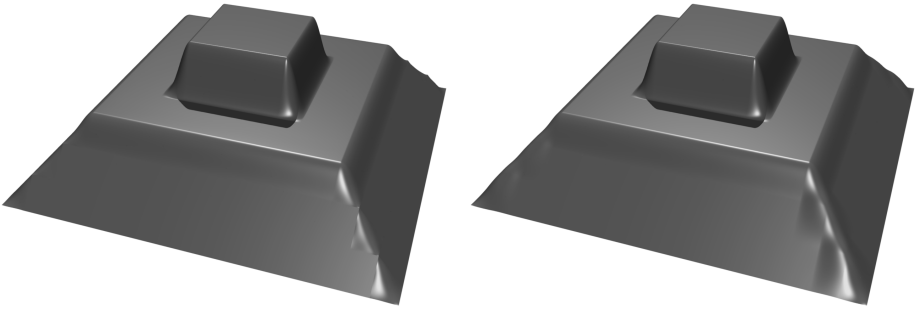

Fig. 1. Test 1: reconstructed surfaces with $\alpha = 3$ and $\alpha = 5$, $h = 1/16$

3.2 Test Cases: Real Terrain Data

Next, we present results for two data sets taken from real terrain data. Test 2 is defined on a 20×20 mesh and it is one of the reference tests in [16]; Test 3

Table 1. Results for test 1 (left), 2, and 3 (right) with $\alpha = 3$

$1/h$	16	32	64	128	256	20×20	100×100
IP iter.	15	16	17	18	19	21	28
$\hat{n}$	2 401	9 409	37 249	148 225	591 361	3 721	90 601
m	10 368	41 728	167 424	670 720	2 684 928	16 240	409 200
PCG iter.	1 512	1 599	2 437	3 628	4 751	4 656	3 923
Ratio×4	—	4.23	6.10	5.95	5.24	—	—
Time, sec.	4.62	21.22	126.68	754.37	3 908.27	21.78	498.55
Ratio	—	4.59	5.97	5.95	5.18	—	—

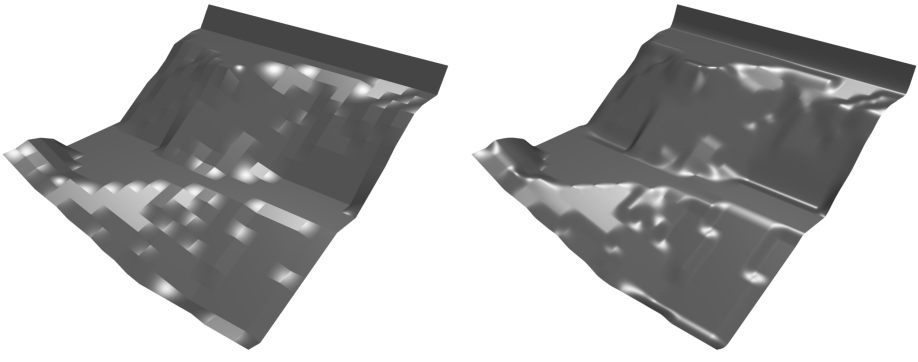

Fig. 2. Test 2: $\mathbb{Q}_1$ interpolant and reconstructed surface with $\alpha = 3$

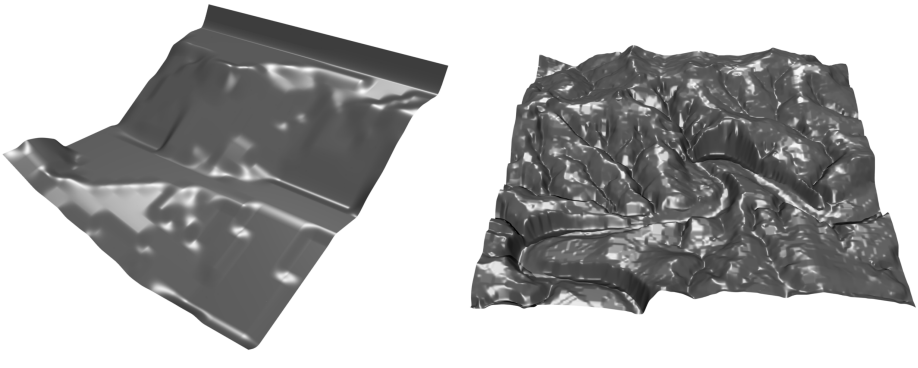

Fig. 3. Test 2, reconstructed with $\alpha = 5$, and Test 3, reconstructed with $\alpha = 3$

is defined on a 100×100 mesh and represents a 3×3 km terrain near Barton Creek in Austin, Texas. The results for test 2 are shown in Figures 2 and 3, and Figure 3 also shows the reconstructed terrain for test 3. In Table 1 (right), we present the computational results for the IP method applied to tests 2 and 3.

4 Conclusion

As claimed by Lavery [5,1], we have observed that the L^1-metric is suitable for reconstructing piecewise smooth data in the sense that it is non-oscillatory. We have proposed a finite element technique which is more flexible than cubic splines. We have proposed a preconditioned interior-point technique whose complexity scales like $n^{5/4}$.

Acknowledgement. This material is based upon work supported by the National Science Foundation grant DMS-0510650.

References

1. Lavery, J.E.: Shape-preserving interpolation of irregular data by bivariate curvature-based cubic L_1 splines in spherical coordinates. Comput. Aided Geom. Design 22(9), 818–837 (2005)
2. Darbon, J., Sigelle, M.: Image restoration with discrete constrained total variation. I. Fast and exact optimization. J. Math. Imaging Vision 26(3), 261–276 (2006)
3. Chambolle, A., Lions, P.-L.: Image recovery via total variation minimization and related problems. Numer. Math. 76(2), 167–188 (1997)
4. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D. 60, 259–268 (1992)
5. Lavery, J.E.: Univariate cubic L_p splines and shape-preserving, multiscale interpolation by univariate cubic L_1 splines. Comput. Aided Geom. Design 17(4), 319–336 (2000)
6. Candes, E.J., Tao, T.: Decoding by linear programming. IEEE Trans. Inform. Theory 51(12), 4203–4215 (2005)
7. Candès, E.J., Romberg, J.K., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. Comm. Pure Appl. Math. 59(8), 1207–1223 (2006)
8. Lavery, J.E.: Solution of steady-state one-dimensional conservation laws by mathematical programming. SIAM J. Numer. Anal. 26(5), 1081–1089 (1989)
9. Lavery, J.E.: Solution of steady-state, two-dimensional conservation laws by mathematical programming. SIAM J. Numer. Anal. 28(1), 141–155 (1991)
10. Guermond, J.L.: A finite element technique for solving first-order PDEs in L^P. SIAM J. Numer. Anal. 42(2), 714–737 (2004) (electronic)
11. Guermond, J.L., Popov, B.: Linear advection with ill-posed boundary conditions via L^1 minimization. Int. J. Numer. Anal. Model. 4(1), 39–47 (2007)
12. Guermond, J.L., Popov, B.: L^1-minimization methods for Hamilton-Jacobi equations: the one-dimensional case. Numer. Math. 109(2), 269–284 (2008)
13. Guermond, J.L., Popov, B.: l^1-minimization methods for Hamilton-Jacobi equations. SIAM J. Numer. Anal. (accepted)
14. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge University Press, Cambridge (2004)
15. Wright, S.J.: Primal-dual interior-point methods. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (1997)
16. Wang, Y., Fang, S.C., Lavery, J.E.: A compressed primal-dual method for generating bivariate cubic L_1 splines. J. Comput. Appl. Math. 201(1), 69–87 (2007)