Fair Polyline Networks for Constrained Smoothing of Digital Terrain Elevation Data

Michael Hofer, Guillermo Sapiro, *Senior Member, IEEE,* and Johannes Wallner

***Abstract*— We present a framework for smoothing grid-like digital terrain elevation data, which achieves fair shape by means of minimizing an energy functional. The minimization is performed under the side-condition of hard constraints which come from available horizontal and vertical accuracy bounds in the standard elevation specification. We introduce the framework and demonstrate the suitability of this method for the tasks of accuracy-constrained smoothing, feature-preserving smoothing, and filling of data voids.**

***Index Terms*— Digital terrain elevation data (DTED), surface smoothing, fair polyline networks, guaranteed error bounds, topography preserving, terrain visualization, terrain compression.**

I. Introduction

SMOOTHING digital terrain elevation data (DTED) with guaranteed error bounds and feature preservation is of great importance in practical geoscience tasks, both for visualization purposes and for post-processing (e.g., compression).

A typical data set we are working with is depicted in Fig. 1. We see that the iso-height contours are often jagged, and false contours appear (small "islands"). This is due in part to the high level of detail present in the data and in part to data acquisition and generation limitations. The noisy aspect of the DTED can also be seen by means of a shaded relief (Fig. 2). After constrained-smoothing with the technique here proposed, iso-height contour lines look like those in Fig. 3.

The data we use have been provided by the U.S. National Geospatial-Intelligence Agency (NGA). The data have been obtained by the Shuttle Radar Topography Mission (SRTM) in February 2000, one of the most significant space surveys of earth ever undertaken, see e.g., *http://www2.jpl.nasa.gov/srtm/* and [1]. SRTM is a joint project between NASA, NGA, the German Aerospace Center (DLR), and the Italian Space Agency (ASI). The data complies to the DTED-2 specification, i.e., is given as a uniform gridded matrix of terrain elevation values with a spacing of 1 arc second. The specification also includes information about absolute errors (or accuracy), both in horizontal as well as in vertical directions.[1] This means that the data includes specific information about the possible errors in the posts positions (circular horizontal error) and the reported height (vertical error). This standard additional accuracy information, which translates into l_∞ norms, is

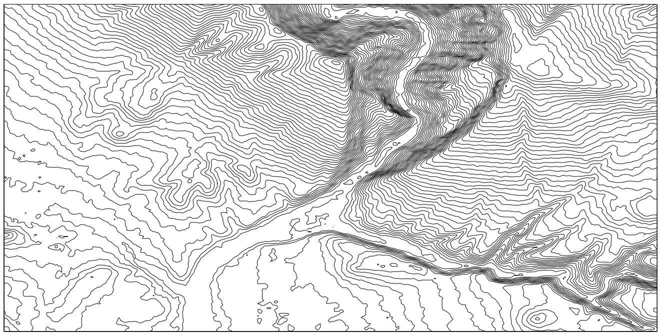

Fig. 1. Iso-height contour plot of original elevation data.

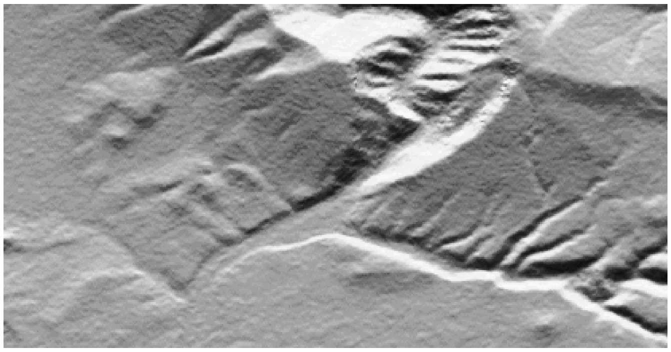

Fig. 2. Shaded relief of original data.

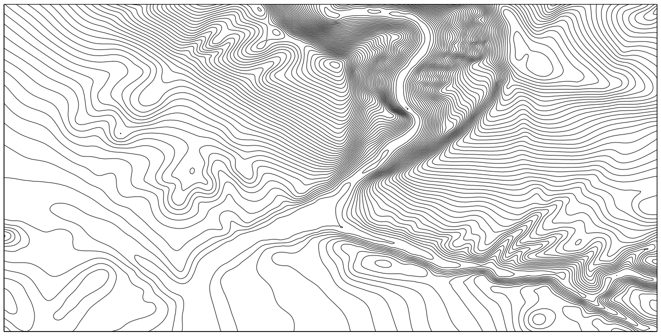

Fig. 3. Iso-height contour plot of smoothed elevation data.

Manuscript received July 21, 2005; revised December 9, 2005. We thank Paul Salamonowicz and Edward Bosch for introducing us to the problem addressed in this paper. This work was supported by the National Geospatial-Intelligence Agency (USA), the Office of Naval Research (USA), the National Science Foundation (USA), the National Institutes of Health (USA), and the Austrian Science Fund (FWF) under grant P15911.

M. Hofer and G. Sapiro are with the Department of Electrical and Computer Engineering, University of Minnesota, 200 Union St. SE, Minneapolis, MN, 55455, USA. Email: hofer@geometrie.tuwien.ac.at; guille@umn.edu. Tel: +1 612 625 1343. Fax: +1 612 625 4583.

J. Wallner is with Vienna University of Technology, Austria. Email: wallner @geometrie.tuwien.ac.at.

[1] For details on the specification, see for example *http://www.fas.org/irp/program/core/dted.htm*. See also *http://mountains.ece.umn.edu/∼guille/dtedspecification.pdf* for a copy of the official DTED specification.

exploited by our proposed smoothing framework.

It is of great importance that smoothing digital elevation data respects the given error bounds. In other words, the smoothed data must be in between the absolute accuracy bounds provided by the specification. This is needed in order to guarantee that the processed data is in the allowed (by the data specification) height and position range. In this paper we present an algorithm which does just that, and is also capable of filling holes as well as preserving features in a consistent manner. The hard error (or accuracy) bounds are represented as tolerance cylinders, where the surface is permitted to move. The smoothing is accomplished by means of an energy minimization technique (fair polyline network), constrained to keep the surface inside the tolerance bounds. Relaxing or limiting the size of the cylinders, void filling and feature preservation are naturally obtained within the same framework. The proposed technique can be applied both to smooth the whole data or single iso-height lines.

The remainder of the paper is organized as follows. In Section II we briefly describe related prior art. Our contributions are put forward in Section III. In Section IV, we first define fair polyline networks and then discuss the error bounds and the smoothing procedure. For validation of the algorithm, we introduce a deviation measure based on a gauge body. Furthermore, we show that besides smoothing, our algorithm is also capable of preserving features and filling voids. In Section V we present experimental results, and in Section VI we conclude the paper.

II. Previous work

The present paper is motivated by our work on variational interpolation of subsets [2], on energy-minimizing splines in manifolds [3], and on fair curve and polyline networks in nonlinear geometries [4]. In the following we review literature related to surface smoothing, guaranteed error bounds, and hole filling, with a focus mainly on the *geometry processing* rather than on the geosciences community.

A. Surface smoothing

For elevation data, the filtering of noise and smoothing of the geometry has been of great interest. Constraints imposed on the smoothing process include the preservation of linear and non-linear surface features such as sharp edges, corners, or non-planar curves. This can be achieved for example using the so-called anisotropic smoothing methods (cf. [5]–[7]). There are only few publications where hard error bounds, as dictated by the standard DTED file specification, are incorporated into these geometric regularization methods. Recently, geometric active contours have been used as the base of a constrained regularization framework for digital elevation data [8].

B. Error bounds and the tolerance aspect

While there are many contributions concerning error propagation in digital elevation models (see for example [9] and the references therein), it is apparently difficult to locate previous work on smoothing of terrain data with guaranteed error bounds (being the only paper to the best of our knowledge, our work in [8]). Thus, it is perhaps no coincidence that the present contribution to this topic has its origin in geometry processing and geometric modeling. We would also like to mention that the topic is related to *tolerance analysis*: It is common to locate imprecisely defined entities in computations by tolerance zones. The most prominent example of this is to compute with intervals instead of with real numbers (see e.g. [10]). Geometric operations with tolerance zones have been studied later [11]. In that paper, energy-minimizing curves which interpolate imprecisely defined points (i.e., tolerance zones) are considered, which is also the topic of [2], and which is closely related to the problem solved in the present paper.

In the approximation theory literature, the problem of regularization with constraints has been addressed by Kimeldorf and Wahba [12]. They showed how to compute one-dimensional splines with hard vertical-error constraints. While this elegant approach can easily be extended to higher dimensions, it does not include the horizontal freedom given by the horizontal absolute error. It is also not developed for the additional geometric constraints that are natural to add in our framework. The theory of *total least squares* [13] also addresses the “freedom of motion” of the given data, both in the vertical and horizontal position. In its original form, although computationally very efficient, the framework does not provide hard constraints (that is, the error is not guaranteed to be below the allowed bounds), neither does it include any kind of explicit regularization or geometric terms. In order to add these important constraints, the problem has to be transformed into a variational formulation, much of the flavor here introduced. The recent work in [14] presents the problem of level-set estimation as a tree optimization one, with guaranteed optimality but no guaranteed error bounds.

C. Void filling

It is possible that in some areas no elevation values are available – we refer to these void areas as ‘holes.’ There are various reasons for holes in the data. They can be caused, e.g., by occlusion or poor signal returns. While our approach fills holes in an intrinsic way via the smoothing process, we still have to provide an initial guess for the missing data. Here any of the known methods for filling voids can be used.

The literature on hole filling algorithms for geometric models roughly fits in three categories: *surface based*, *volumetric*, and *example based* methods. The article [15] surveys the literature on the first two approaches up to the year 2002. Ideas from image inpainting are used in the volumetric approach by [16]. Filling holes in point set surfaces is also discussed in [17]. Recently, example based methods have emerged [18]–[20]. It is interesting to note that the example based approach is most closely related to a common practice of filling voids in elevation data: One uses data from alternate sources and blends them with the data surrounding the void (the ‘fill & feather method,’ cf. [21], [22]).

Extensive literature exists on the interpolation of data from level-lines, e.g., [23], [24]. This is a slightly different problem, as holes are often significantly larger than the common distance between level-lines.

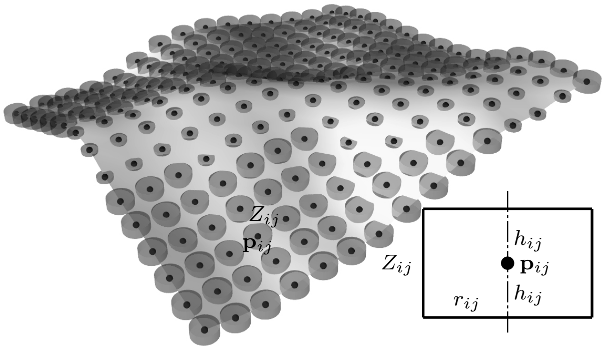

Fig. 4. An initial surface interpolating the given data points $\mathbf{p}_{ij}$. The surface can be smoothed such that it stays within all tolerance cylinders Z_{ij} of radius r_{ij} and height $2h_{ij}$. Both radius and height are specified with the DTED information. A different size of the error cylinders can be used to preserve surface features (e.g., ridges) during the smoothing.

III. Contributions of the present paper

Elevation data of the DTED-2 specification are given as a height field over a uniform grid in the xy-plane, such that every data point has horizontal and vertical error (accuracy) bounds. This error translates into a hard, l_∞, bound. Fig. 4 shows data points $\mathbf{p}_{ij}$ together with their error bounds in the form of cylinders of revolution Z_{ij}, and an initial surface that interpolates the given points $\mathbf{p}_{ij}$. The objective is to compute a smooth surface that stays within the given error margins dictated by the cylinders.

The framework we propose views the height field over the xy-plane as a network consisting of x-parallel and y-parallel polylines, as shown by Fig. 5. We perform smoothing by minimizing a discrete bending energy of these polylines, always respecting the error bounds. Our framework is very flexible and allows e.g., to use individual error bounds for every data point, fill voids in the data, or smooth only a subset of the data for feature preservation. We later illustrate this at hand of several examples.

IV. Fair polyline networks for terrain smoothing with guaranteed error bounds

We first define fair polyline networks and then we show how to use them for smoothing of digital terrain elevation data with guaranteed error bounds.

A. Fair polyline networks

Our smoothing procedure is based on minimizing the energy of a polyline network. We first define the energy of a single polyline, and then the energy of the polyline network as the sum of energies of all the polylines that contribute to the polyline network. For a smooth curve $\mathbf{c}(t)$, defined in some parameter interval $[a, b]$, the linearized bending energy (cubic spline energy) is defined by $E(\mathbf{c}) = \int_a^b \|\ddot{c}(t)\|^2 dt$. Here $\ddot{c}$ denotes the second derivative vector of the curve $c(t)$ with respect to the curve parameter t. Its minimizers under interpolation conditions are the cubic B-spline curves [25]. A *polyline* $\mathbf{p} = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_L)$, as a discrete curve, possesses a discrete linearized bending energy:

$$E = \sum\nolimits_{i=2}^{L-1} \|\Delta^2 \mathbf{q}_i\|^2, \quad \Delta^2 \mathbf{q}_i = \mathbf{q}_{i-1} - 2\mathbf{q}_i + \mathbf{q}_{i+1}. \quad (1)$$

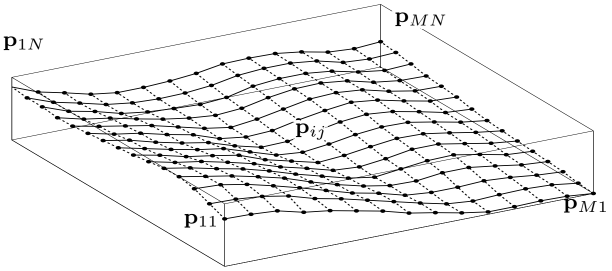

Fig. 5. A polyline network with vertices $\mathbf{p}_{ij}$ and two families of polylines (solid and dotted) representing a height field.

Curve networks and polyline networks, which are the topic of [4], have energies which are defined as the sum of the energies of the curves or polylines which they are made of. The given elevation data constitute a rectangular array of points: $\mathbf{p}_{ij} = (x_{ij}, y_{ij}, z_{ij})$, $(i = 1, \ldots, M,\ j = 1, \ldots, N)$. We define the energy of the data collection to be the sum of energies of the N different polylines defined by $j =$ const. and the M different polylines defined by $i =$ const.:

$$\begin{aligned} E = &\sum\nolimits_{j=1}^{N} \sum\nolimits_{i=2}^{M-1} \|\mathbf{p}_{i-1,j} - 2\mathbf{p}_{i,j} + \mathbf{p}_{i+1,j}\|^2 \\ &+ \sum\nolimits_{i=1}^{M} \sum\nolimits_{j=2}^{N-1} \|\mathbf{p}_{i,j-1} - 2\mathbf{p}_{i,j} + \mathbf{p}_{i,j+1}\|^2. \quad (2) \end{aligned}$$

A *fair* polyline network is one which has minimal energy among all networks which fulfill a fixed set of constraints.

B. Error bounds

The data sets we use consist of points $\mathbf{p}_{ij}$ which are equally spaced in x and y direction. We might, without loss of generality, assume that $x_{ij} = g_x \cdot i$ and $y_{ij} = g_y \cdot j$, where g_x, g_y specify the grid element size (1 arc second for DTED-2). The data points contain errors, both in horizontal and vertical direction. If the horizontal error of a data point $\mathbf{p}_{ij}$ is bounded by r_{ij}, and the vertical error by h_{ij}, then the true location of that data point is within a cylinder Z_{ij} of diameter $2r_{ij}$ and height $2h_{ij}$, which is centered in the given point $\mathbf{p}_{ij} = (x_{ij}, y_{ij}, z_{ij})$.[2] Thus each point $\mathbf{p}_{ij}$ is equipped with its own *tolerance cylinder* Z_{ij} (Fig. 6, left). We also refer to these tolerance cylinders as *error cylinders*. They are hard constraints, which means that the terrain surface we are seeking has to pass through all Z_{ij}'s.

Fig. 6, top right, shows the initial state of the network, only one polyline is shown as a representative of the surface defined by the data. Our goal ultimately is to move the points $\mathbf{p}_{ij}$ such that the energy (2) becomes smaller, but the surface still passes through the cylinders Z_{ij}. We can achieve this, e.g., by requiring that the points $\mathbf{p}_{ij}$ themselves do not leave Z_{ij}, as illustrated by Fig. 6, at right, center. This condition, which is referred to as *Option 1*, however, is stricter than actually necessary. It is certainly sufficient that for each vertex $\mathbf{p}_{ij}$, one of the two polylines meeting there meets Z_{ij} (Fig. 6, bottom right). This slightly more relaxed condition is referred to as *Option 2*.

[2] Other error/accuracy models will just define different tolerance shapes, without altering the framework here presented. Cylinders are the appropriate shape to represent the absolute errors specified with DTED.

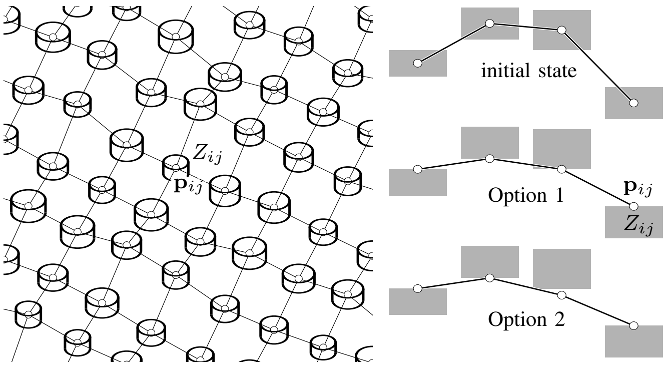

Fig. 6. Smoothing height fields via fair polyline networks. (Left) Polyline network with the tolerance cylinders Z_{ij} associated with the vertices $\mathbf{p}_{ij}$. (Right) At top, initial state before optimization. At center, Option 1 of optimization (all vertices stay inside the tolerance cylinders). At bottom, Option 2: The vertices are allowed to leave the cylinders, provided the surface still passes through them.

C. The smoothing procedure

Minimizing the quadratic function (2) of the variables x_{ij}, y_{ij}, z_{ij}, subject to the constraints mentioned above, is a quadratic programming problem with non-convex side conditions. It contains too many variables for just submitting it to a generic optimization procedure as, e.g., provided by mathematical software. The following properties however allow a more direct approach:

(a) If computed with x and y coordinates of the initial data alone, the energy would be zero, owing to the regular alignment of initial data. We therefore do not expect the points $\mathbf{p}_{ij}$ to move very much in x or y direction during optimization. Indeed this is confirmed by numerical experiments.

(b) At each location, the smooth, energy-minimizing terrain appears to be defined by nearby input data. Thus there is no need for globally minimizing the energy over large data sets.

(c) The condition that $\mathbf{p}_{ij}$ remains inside Z_{ij} (Option 1) leads to a convex optimization problem which has a unique solution. As illustrated by Fig. 6 (right), the difference between Option 1 and Option 2 is not very big. This means that the optimization problem we have to solve when using Option 2 is 'convex enough' so that we do not expect a direct minimization procedure getting stuck in a local minimum.

In view of property (a), we only use the z coordinates of the data points as variables for minimization:

$$\begin{aligned} E = &\sum\nolimits_{j=1}^{N} \sum\nolimits_{i=2}^{M-1} (z_{i-1,j} - 2z_{i,j} + z_{i+1,j})^2 \\ &+ \sum\nolimits_{i=1}^{M} \sum\nolimits_{j=2}^{N-1} (z_{i,j-1} - 2z_{i,j} + z_{i,j+1})^2. \end{aligned} \tag{3}$$

Because of (c), we employ a gradient descent method, with the original elevation data as initial condition. It is elementary that the gradient of the energy is given by

$$\begin{aligned} (\nabla E)_{ij} = &(z_{i-2,j} + z_{i+2,j} + z_{i,j-2} + z_{i,j+2}) \\ &- 4(z_{i-1,j} + z_{i+1,j} + z_{i,j-1} + z_{i,j+1}) + 12 z_{i,j}, \end{aligned} \tag{4}$$

provided $i, j > 2$ and $i < M-1$, $j < N-1$ (that is, we superimpose the masks $[1, -4, 6, -4, 1]$ in both x and y direction). As we approach the boundary, the standard mask becomes $[-2, 5, -4, 1]$ and finally $[1, -2, 1]$.

Optimization is basically implemented as follows: First, we find a direction of descent, e.g., by letting $g_{ij} := -(\nabla E)_{ij}$. We consider the 1-parameter variation E_t of the energy (3) defined by the z coordinates $z_{ij}(t) = z_{ij} + t \cdot g_{ij}$. The dependence of E on t is quadratic, so it is easy to find a parameter $t = t_0$ where E_t has a minimum. We replace z_{ij} by $z_{ij} + t_0 \cdot g_{ij}$, but vertices which have moved too far (violating the constraints) are pulled back. This procedure is iterated. Actually the procedure described here is rather unsophisticated. Improvements are the following: (i) If $\mathbf{p}_{ij}$ is already in a position where it must not move higher because of the boundary conditions, but $g_{ij} > 0$, we let $g_{ij} = 0$. (ii) the same with 'lower' instead of higher. (iii) As is well known in multivariate optimization, the direct gradient descent method is usually not very efficient. We use a conjugate gradient method for updating the direction of descent in each step. (iv) In view of the large number of variables, we use a multigrid method for minimization.

As has already been mentioned, Option 1 leads to a convex optimization problem, whereas Option 2 leads to a non-convex one. Numerical experiments however show that no problems with local minima occur, so the amount of non-convexity present seems to be low. For numerical optimization in general, see e.g., [26]. Further note that when minimizing (3) with Option 1, the horizontal position error does not play a role at all.

D. Measuring the deviation

In order to measure the deviation of the smoothed surface, defined by vertices $\mathbf{q}_{ij}$, from the original data $\mathbf{p}_{ij}$, we use the tolerance cylinders Z_{ij} as gauge bodies. When scaling Z_{ij} with center $\mathbf{p}_{ij}$, there is a smallest factor λ_{ij} such that the scaled cylinder touches the surface defined by the smoothed data (which means that the scaled cylinder touches either the x-parallel or the y-parallel polylines defined by the smoothed data). This factor λ_{ij} equals zero, if the vertex $\mathbf{p}_{ij}$ has not moved at all, i.e., if $\mathbf{q}_{ij} = \mathbf{p}_{ij}$. If $\lambda_{ij} \leq 1$, then the surface passes through Z_{ij}.

E. Void filling

Missing data may be treated in different ways. There might be a procedure for filling voids which is appropriate for a certain application and which is applied before fair polylines are used for smoothing. It is however worth noting that fair polylines are also capable of filling voids. If z_{ij} is unknown, we first attempt a crude guess at it (any method for filling voids is sufficient for that), and endow the vertex $\mathbf{p}_{ij}$ with a large tolerance cylinder Z_{ij}, so that the vertex can move (more) freely during optimization. Hole filling then is an intrinsic component of the algorithm.

V. Experimental results

All data sets described below are conforming to the DTED-2 standard, with a post spacing of 1 arc second. The horizontal and vertical error margins are 13 and 5 meters, resp., for all vertices. This means that after smoothing the surface must pass through vertically oriented tolerance cylinders of base diameter 26 m and height 10 m centered in the individual data points. The size of those cylinders is the same for all points, except for those where no height data are available (they can move freely) or whose movement is restricted on purpose (Examples 3 and 4).

A. *Example 1*

This data set has a size of 800×600 posts and is shown in Fig. 7 as a shaded relief. It represents an area of approximately 24×18 km. The original data set has elevation values between $z_{\min}$ =339 m and $z_{\max} = 1103$ m, and the smoothed data set has elevation values between $z_{\min} = 337.5$ m and $z_{\max} = 1098$ m. Details of this example have already been used for illustration in Figures 1, 2, and 3.

In order to analyze the asymptotic behavior of the energy minimization, we run much more iterations than necessary in actual applications. The behavior of the network energy using smoothing Option 2 is:

# iterations	0	10	100	1000
Energy	4817	727	628	623

The computational cost of one round of iteration is roughly proportional to the amount of input data. Fig. 8 shows a histogram of vertex movements measured by means of the tolerance cylinders as gauge bodies (cf. Section IV-D). The number of points whose movement is a certain percentage of the maximum possible movement is decreasing from left to right, as only to be expected. It is interesting to note that a substantial number of points actually move as far as possible in order to minimize energy. This is clearly seen by the height of the bar at the extreme right.

Fig. 9 illustrates the movement of vertices during the smoothing process. The height difference in meters is coded as a grey value. The dark lines show valleys, which during smoothing are lifted, whereas the light areas show ridges, where the smooth terrain is lower than the original one. In both cases, the deviation of the smoothed terrain from the original one is always consistent with the DTED-2 specified error/accuracy bounds, and can be further controlled by changing the cylinder sizes, see below.

B. *Example 2: Options comparison*

Our second example demonstrates the difference between the two options mentioned in Section IV-B. Option 1 keeps the data points inside the tolerance cylinders, whereas Option 2 ensures that the surface passes through the tolerance cylinders, but allows the vertices $\mathbf{p}_{ij}$ to leave the respective cylinders Z_{ij}. The results of smoothing are visualized in Fig. 11. Minimal elevation values for the original data set, and the two smoothed data sets are 431.00 m, 427.28 m, and 426.00 m, respectively. Maximum elevation values are 724.00 m, 723.27 m, and 723.27 m.

While the general effect of smoothing is clearly visible in the contour plots for both options (see Fig. 11(a)–(c)), it is hard to see any difference between options 1 and 2. The effect that vertices may move outside their tolerance cylinder

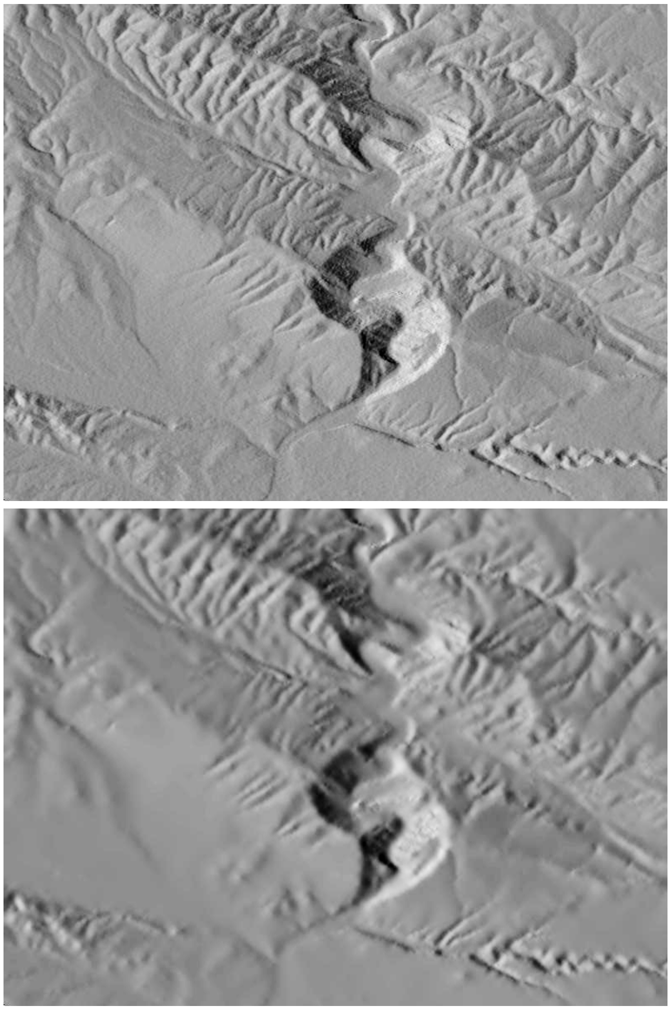

Fig. 7. Data set of Example 1 before (top) and after (bottom) smoothing.

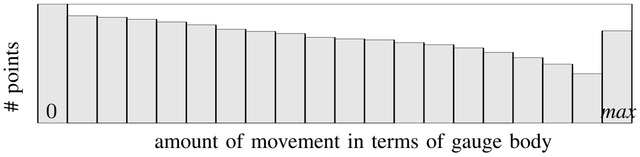

Fig. 8. The vertex movement histogram mentioned in Example 1.

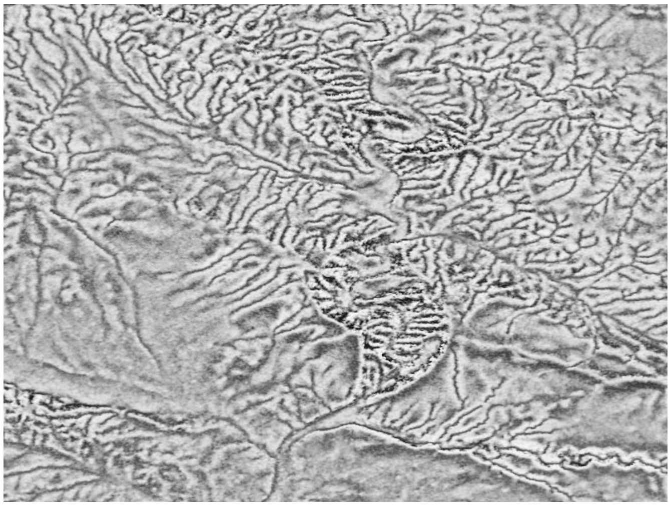

Fig. 9. Vertex movement visualization for Example 1.

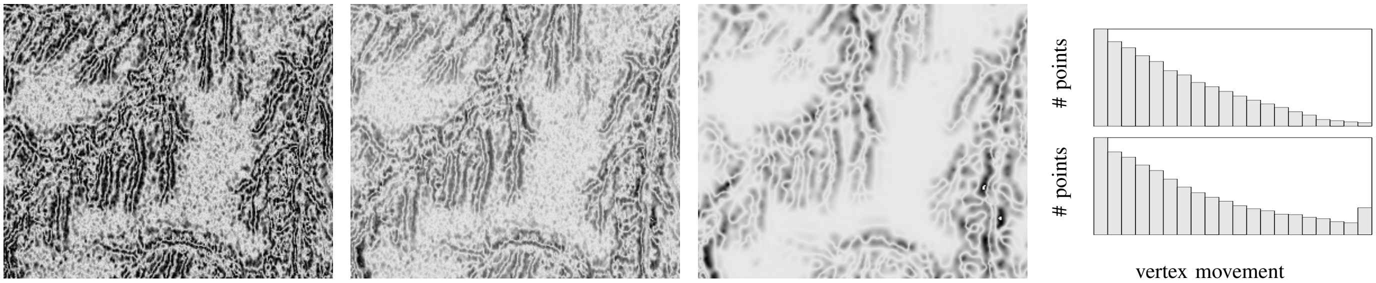

Fig. 10. The data set of Example 2. From left: (a) Deviation of smoothed data (Option 1) from original; (b) Deviation of smoothed data (Option 2) from original; (c) Difference between two options of smoothing; (d) Vertex movement histogram (Top: Option 1, Bottom: Option 2.).

is strongest where the terrain is steep. In flatter areas, the vertices cannot move away much from the boundary of the tolerance cylinder. This means that the maximum deviation of the smoothed terrain from the original occurs in those areas where the terrain is steep. In this example, this happens only in small parts of the data set. Figures 10(a) and 10(b) illustrate this deviation, where the absolute value of deviation is encoded in grey values: Maximum deviation is black, minimum is white.

The fact that Option 2 causes deviation peaks in small areas causes Fig. 10(b) to look lighter than Fig. 10(a), even if the absolute deviation values are greater in general. Fig. 10(c) shows the difference between smoothing with Option 1 and smoothing with Option 2: the absolute value of the height difference is encoded in grey values (black is maximal deviation). The vertex movement statistics of Fig. 10(d) do not show absolute height data, but the deviation computed with the tolerance cylinders as gauge bodies.

C. Example 3: Hole filling and feature preservation

The third example uses a small data set of size 217×181, which contains several holes (Fig. 12).

Before smoothing, the missing z coordinates are guessed by a simple averaging procedure (any of the known procedures for filling voids could be used). During smoothing, they are allowed to move freely. This example illustrates the fact that fair polyline networks are capable of filling voids in one pass together with constrained smoothing (Fig. 12(b)).

We are now going to further exploit the great amount of flexibility which is present in our way of treating constraints. For instance, features of the terrain which should survive the smoothing process with greater accuracy than other parts of the data set can be given smaller tolerance cylinders. An example of this is given by Fig. 12(c), where smoothing has been performed with three different cylinder sizes (diameter/height in meters): 26/10 for most points, 2/1 for the feature marked in grey, and an infinite size for voids in the data.

The difference between the two different ways of smoothing is shown by Fig. 13. Here absolute values of z coordinate differences are encoded as grey values, with white as minimum and black as maximum. Data voids are also shown in black. The small movement of vertices in the feature area shows up as white. A shaded relief of absolute value of the difference of the two smoothed data sets is shown in Fig. 13(c).

Features which are to preserved can either be marked by hand or detected automatically, which is a nontrivial task in itself, although has been extensively studied in the literature (see e.g. [27]). Actually images like the vertex movement visualization of Example 1 (Fig. 9) can aid the detection of features.

D. Example 4: Individual contour smoothing

When working with digital terrain elevation data, one might want to be able to perform the following two tasks: (a) the whole data set is smoothed with guaranteed error bounds. On demand a contour plot of the smoothed terrain is created. (b) a single smooth contour line at a certain fixed elevation $z = H$ is extracted from the original terrain data, always respecting the given error bounds. The first task can be performed by the algorithm described above and is shown by Examples 1–3. The second task can be done by the same algorithm which acts on a subset of the given data defined by the inequality $H_1 < z < H_2$, where $[H_1, H_2]$ is an interval which contains the height value H under consideration and is wide enough to include the tolerance cylinders.

Figures 14 and 15 illustrate this procedure: We use a data set of size 436×263 containing both mountainous and flat regions between $z_{\min} = 1894\,\mathrm{m}$ and $z_{\max} = 3033\,\mathrm{m}$. We pick the $2212\,\mathrm{m}$ contour for smoothing. The region $H_1 < z < H_2$ with $H_1 = 2192\,\mathrm{m}$ and $H_2 = 2232\,\mathrm{m}$ is shown in Fig. 14(c). The result of smoothing is shown in Fig. 15(a). For comparison we have also smoothed the whole data set and show its corresponding $2212\,\mathrm{m}$ contour in Fig. 15(b). In Fig. 15(c) we show a shaded relief of the deviation from the original data when smoothing only a neighborhood of a single contour.

E. Compression results

For the sake of demonstration, we have applied the standard JPEG-LS lossless "*loco16e*" compression algorithm [28], which is available at *http://www.hpl.hp.com/loco/locodown.htm*, to the datasets of Examples 1–4, saved in PGM format. JPEG-LS is the only compression standard compatible with hard error constraints [29]. Compression rates before and after smoothing were 3.87:1 / 3.54:1 / 2.69:1 / 3.44:1 and 4.99:1 / 5.20:1 / 3.57:1 / 4.86:1, respectively. This is an improvement of 29%, 47%, 33%, and 41%, respectively.

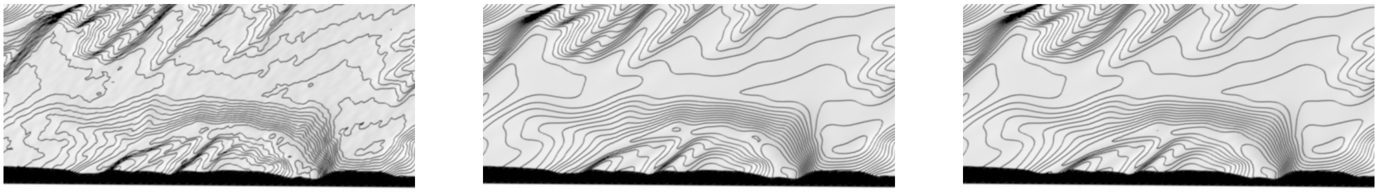

Fig. 11. The data set of Example 2 (detail). From left to right: (a) Original data. (b) Smoothing with Option 1. (c) Smoothing with Option 2.

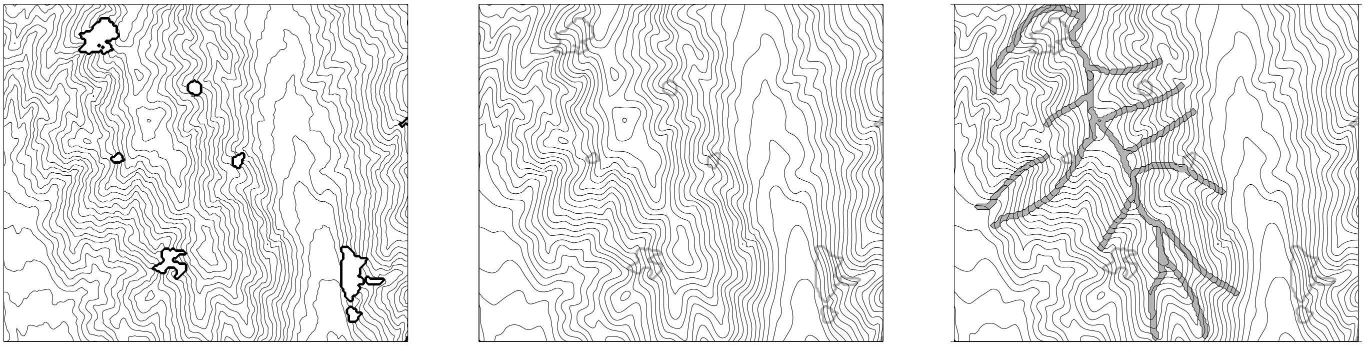

Fig. 12. The data set of Example 3. From left: (a) Original data. (b) After filling voids and smoothing. (c) After filling voids and smoothing, with features.

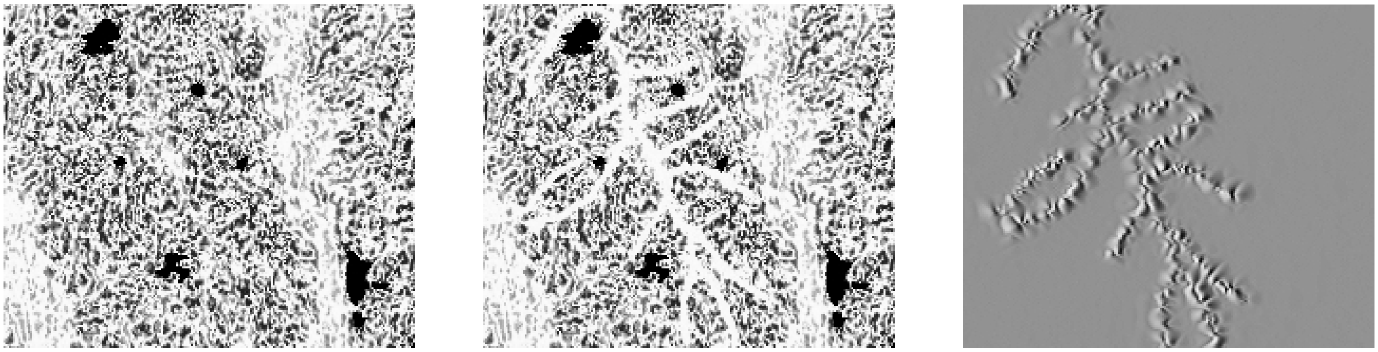

Fig. 13. The data set of Example 3. From left: (a) Height differences between original and smoothed data. (b) Height differences between original and smoothed data, with features preserved. (c) Difference between the two smoothed data sets (with and without features preserved).

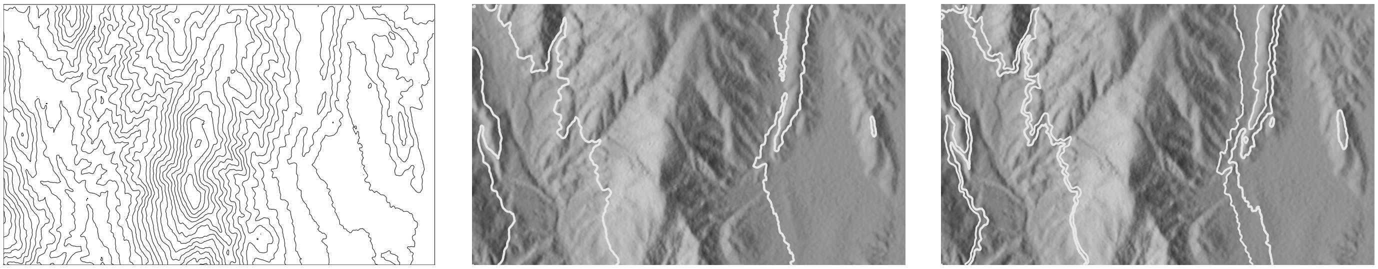

Fig. 14. The data set of Example 4. From left: (a) Contour lines. (b) Shaded relief with single contour (in white) at $z = 2212$ m. (c) A neighborhood of that contour defined by contour lines at 2192 m and 2232 m.

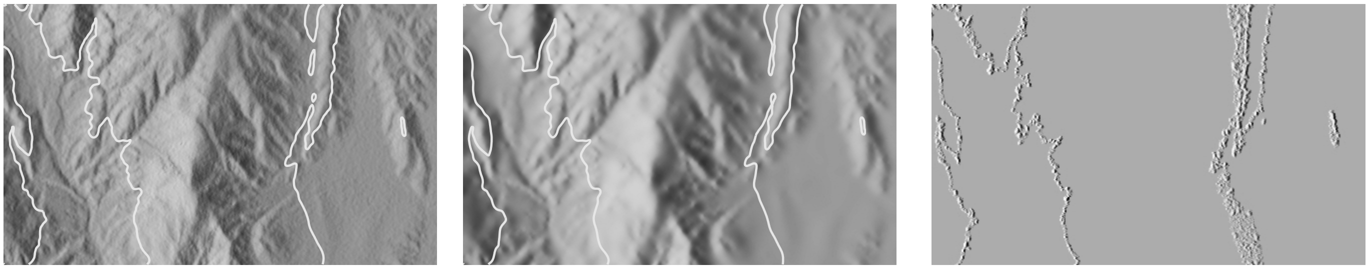

Fig. 15. The data set of Example 4. From left: (a) Smoothing the 2212 m contour by smoothing a neighborhood. (b) Smoothed whole data set with its corresponding 2212 m contour. (c) Deviation from original data when smoothing a neighborhood of a single contour (shaded relief).

VI. Conclusion

We have presented a framework which uses fair polyline networks for smoothing digital terrain elevation data with guaranteed error bounds and feature preservation. The algorithm is capable of smoothing the terrain data with tolerance cylinders of different sizes. These flexible tolerances have two advantages in particular: (i) we can preserve features present in the data by reducing the size of the tolerance cylinders in feature areas, (ii) the algorithm can be used to fill holes present in the original data during the smoothing process. Single contour lines are smoothed via processing of a small neighborhood of that contour line. Another important point we would like to make is that the smoothing approach here presented easily generalizes from height fields to more general surfaces, which are becoming increasingly important in photogrammetry and remote sensing [30].

References

[1] B. Rabus, M. Eineder, A. Roth, and R. Bamler, "The shuttle radar topography mission — a new class of digital elevation models acquired by spaceborne radar," *ISPRS J. Photogrammetry & Remote Sensing*, vol. 57, pp. 241–262, 2002.

[2] J. Wallner and H. Pottmann, "Variational interpolation of subsets," *Constr. Approx.*, vol. 20, pp. 233–248, 2004.

[3] M. Hofer and H. Pottmann, "Energy-minimizing splines in manifolds," *ACM Trans. Graphics*, vol. 23, no. 3, pp. 284–293, 2004.

[4] J. Wallner, H. Pottmann, and M. Hofer, "Fair webs," TU Wien, Geometry Preprint 134, January 2005, ACM SIGGRAPH 2005 Conference Abstracts and Applications, August 2005.

[5] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Anisotropic feature-preserving denoising of height fields and bivariate data," in *Graphics Interface*, M. McCool, Ed. A K Peters, 2000, pp. 145–152.

[6] K. Hildebrandt and K. Polthier, "Anisotropic filtering of non-linear surface features," *Computer Graphics Forum*, vol. 23, no. 3, pp. 391–400, 2004, Proc. Eurographics 2004.

[7] T. Tasdizen and R. Whitaker, "Feature preserving variational smoothing of terrain data," in *IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, O. Faugeras and N. Paragios, Eds., 2003.

[8] A. Mohan, A. Bartesaghi, and G. Sapiro, "Constrained regularization of digital terrain elevation data," Univ. Minnesota, IMA Preprint 2049, 2005, available from *http://www.ima.umn.edu*.

[9] J. Oksanen and T. Sarjakoski, "Error propagation of DEM-based surface derivatives," *Computers & Geosciences*, vol. 31, no. 8, pp. 1015–1027, 2005.

[10] S. L. Abrams *et al.*, "Efficient and reliable methods for rounded-interval arithmetic," *CAD*, vol. 30, pp. 657–665, 1998.

[11] J. Wallner, R. Krasauskas, and H. Pottmann, "Error propagation in geometric constructions," *CAD*, vol. 32, pp. 631–641, 2000.

[12] G. S. Kimeldorf and G. Wahba, "Some results on Tchebycheffian spline functions," *J. Math. Anal. Applic.*, vol. 33, pp. 82–95, 1971.

[13] G. Golub and C. V. Loan, *Matrix Computations*. The Johns Hopkins University Press, 1996.

[14] R. Willett and R. Nowak, "Level set estimation via trees," in *Proc. Int. Conf. Acoustics, Speech & Signal Proc.*, vol. 5, 2005, pp. 1085–1088.

[15] J. Davis, S. R. Marschner, M. Garr, and M. Levoy, "Filling holes in complex surfaces using volumetric diffusion," in *1st Int. Symp. 3D Data Processing, Visualization, and Transmission*. IEEE CS Press, 2002.

[16] J. Verdera, V. Caselles, M. Bertalmio, and G. Sapiro, "Inpainting surface holes," in *Int. Conference on Image Processing (ICIP 2003)*, vol. 2. IEEE, 2003, pp. 903–906.

[17] T. Weyrich, M. Pauly, R. Keiser, S. Heinzle, S. Scandella, and M. Gross, "Post-processing of scanned 3D surface data," *Symposium on Point-Based Graphics*, pp. 85–94, 2004.

[18] A. Sharf, M. Alexa, and D. Cohen-Or, "Context-based surface completion," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 878–887, 2004.

[19] M. Pauly *et al.*, "Example-based 3D scan completion," in *Proc. Symp. Geom. Processing*, M. Desbrun and H. Pottmann, Eds. Eurographics Assoc., 2005, pp. 23–32.

[20] G. H. Bendels, R. Schnabel, and R. Klein, "Fragment-based surface inpainting," in *SGP 2005 – Poster Proceedings*, M. Desbrun and H. Pottmann, Eds. Eurographics Assoc., 2005.

[21] S. Dowding, T. Kuuskivi, and X. Li, "Void fill of SRTM elevation data — principles, processes and performance," in *ASPRS Images to Decision: Remote Sensing Foundation for GIS Applications*, 2004, ASPRS Fall Conference, held Sept. 12–16, Kansas City.

[22] T. Kuuskivi, J. Lock, X. Li, S. Dowding, and B. Mercer, "Void fill of SRTM elevation data: Performance evaluations," in *ASPRS Annual Conference. Geospatial Goes Global: From your neighborhood to the whole planet*, 2005, held March 7-11 in Baltimore. ISBN 1570830762.

[23] P. Soille, "Spatial distributions from contour lines: an efficient methodology based on distance transformations," *J. Visual Communication and Image Representation*, vol. 2, no. 2, pp. 138–150, June 1991.

[24] A. Almansa, F. Cao, Y. Gousseau, and B. Rouge, "Interpolation of digital elevation models using AMLE and related methods," *IEEE Trans. Geoscience and Remote Sensing*, vol. 40, no. 2, pp. 314–325, 2002.

[25] G. Farin, *Curves and Surfaces for CAGD*. Morgan Kaufmann, 2002.

[26] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.

[27] S. Yoshizawa, A. Belyaev, and H.-P. Seidel, "Fast and robust detection of crest lines on meshes," in *Proc. SPM '05*. ACM Press, pp. 227–232.

[28] M. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," *IEEE Trans. Image Processing*, vol. 9, pp. 1309–1324, 2000.

[29] S. Rane and G. Sapiro, "Evaluation of JPEG-LS, the new lossless and controlled-lossy still image compression standard, for compression of high-resolution elevation data," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 39, pp. 2298–2306, 2001.

[30] N. Pfeifer, "A subdivision algorithm for smooth 3D terrain models," *ISPRS J. Photogrammetry & Remote Sensing*, vol. 59, pp. 115–127, 2005.

Michael Hofer is a Postdoc at the University of Minnesota. He received his Ph.D. in Mathematics in 2005 from Vienna University of Technology. His research interests include geometry, geometric modeling and processing, computer aided geometric design, computer graphics and computer vision. M. Hofer was awarded the Best Ph.D. Dissertation Award from the Austrian Mathematical Society in 2005. For more information and a list of publications see the webpage *http://www.geometrie.tuwien.ac.at/hofer*.

Guillermo Sapiro (M'95) currently is with the Department of Electrical and Computer Engineering at the University of Minnesota, where he holds the position of Distinguished McKnight University Professor. He works on differential geometry and geometric partial differential equations, both in theory and applications in computer vision, computer graphics, medical imaging, and image analysis. He has authored and co-authored numerous papers in this area and has written a book published by Cambridge University Press, January 2001.

Johannes Wallner is a member of the Institute of Discrete Mathematics and Geometry at TU Wien in Vienna. He received his Ph.D. from TU Wien in 1997 and researches in geometry. He coauthored the monograph *Computational Line Geometry* published by Springer in 2001. A recent focus of his work is smoothness analysis of nonlinear subdivision schemes, tolerance analysis, and geometry processing. See also the web pages at *http://www.geometrie.tuwien.ac.at/wallner*.