



Technical Section

Sketch-based free-form shape modelling with a fast and stable numerical engine

Yong-Jin Liu^a, Kai Tang^{b,*}, Ajay Joneja^a

^a*Department of Industrial Engineering and Logistics Management, Hong Kong University of Science and Technology, Hong Kong, PR China*

^b*Department of Mechanical Engineering, Hong Kong University of Science and Technology, Hong Kong, PR China*

Abstract

In this paper, we present a feature-based free-form shape modelling technique based on solving a fundamental problem of reconstructing the depth information from 2D sketch planes. First, to mathematically define the problem with the human perception, the proposed technique (1) formulates the 2D shaded regions on sketches by a hybrid thin plate surface model that can exhibit controlled continuity over the recovered 3D surface and (2) formulates the 1D salient open free-form curves and salient corners as linear sketch constraints. The 3D free-form shape from sketch planes is then achieved by solving a linearly constrained quadratic optimization problem which unifies both 2D region-based and 1D contour-based shape information over 2D sketches. Secondly, to solve the formulated optimization problem with an interactive-rate performance, a fast and stable numerical engine is proposed with a rigorous proof that for our specially formulated problem, system decomposition with reduced computational cost is always possible. Stability, accuracy and efficiency are studied in depth for the proposed numerical engine. Finally a prototype system utilizing the proposed technique is presented with two applications that demonstrate the usefulness and effectiveness of the proposed technique.

© 2005 Elsevier Ltd. All rights reserved.

Keywords: Free form shape modelling; Feature modelling; 2D sketch; Numerical computation

1. Introduction

In traditional media such as images and videos, three-dimensional (3D) information of the free-form object is encoded in two-dimensional (2D) information due to the projection on the 2D Euclidean planes. The 2D plane is also the most convenient platform for interactive free-form shape design through sketching by artists. Therefore, recovering the depth information from 2D planes

can find rich applications in *computer vision and computer graphics* (CV/CG) and has been considered as a fundamental problem in CV/CG [1–4]. Many automatic approaches to this problem are known [3], e.g., the shape from X techniques (X stands for *stereo*, *shading*, *texture*, *focus/defocus*, *optical flow*, etc.) However, each of these automatic approaches has its own special application environment with various assumptions (or limitations).

In this paper we consider the general case that scarcely mild information is provided as input. An example is illustrated in Fig. 1. Obviously, fully automatical techniques such as shape from shading would fail in such situations. To offer a practical solution to this

*Corresponding author. Tel.: +852 2358656;
fax: +852 23581543.

E-mail addresses: liuyj@ust.hk (Y.-J. Liu),
mektang@ust.hk (K. Tang), joneja@ust.hk (A. Joneja).

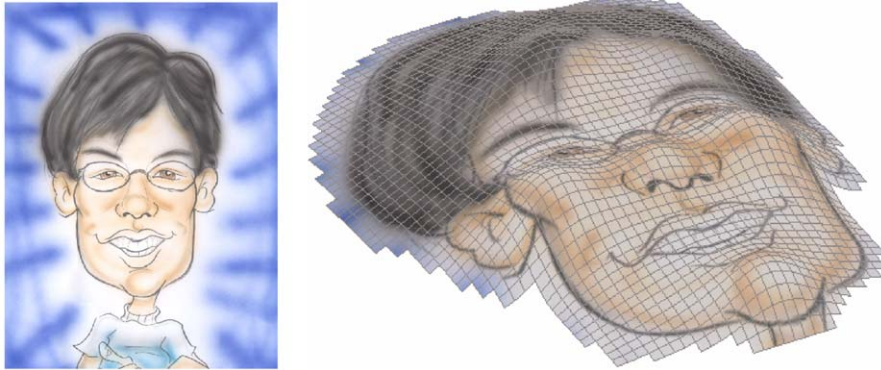


Fig. 1. 2D sketch-based shape modelling.

dilemma, we propose an interactive technique with an optimal real-time performance. The proposed technique can be used in diverse CV/CG applications, which varied from variational visible surface modelling [4,5], texture mapping [6,7], image repair/inpainting [1,2], to single-image-based modelling and rendering [8,9]. In Section 6, two of these applications with examples are presented to demonstrate the usefulness and effectiveness of the proposed technique.

To develop a universal technique with user intervention, the fundamental depth-recovery problem can be generally stated as follows.

Depth-recovery problem: Given a set of user-specified, arbitrarily located, parameterized features $(x_i(t), y_i(t), F_i(t))$, where $(x_i, y_i) \in \Omega \subset \mathcal{R}^2$, find a smooth Monge patch $\mathcal{M}(x, y) = (x, y, z(x, y)) \subset \mathcal{R}^3$ with variously controlled continuities, which takes the given properties F_i at (x_i, y_i) .

1.1. Related work

The general statement of the depth-recovery problem can be regarded as a variation of the *scattered data modelling problem* [10], whereas most classical scattered data interpolating techniques found piecewise continuous functions, splines in particular, fit to the scattered data [11]. From practical point of view, nowadays, discrete representation with values¹ only at regular grids is widely used for shape description with computers [12]; we note that recently an elegant computational model using the language of geometric algebra is developed for computer graphics applications with polygonal meshes [13]. In this paper we adopt a quadrilateral mesh for shape description and deduce a fast numerical solution to the depth-recovery problem that explicitly avoids computing any continuous function; instead, only values at regular grids are recovered.

¹Here *value* includes not only depth information but also other scalar properties.

The reconstruction of surface depth values from a planar region with arbitrarily located (i.e., frequently non-uniformed) interpolating data has been shown in [14] to be an ill-posed problem in terms of solution existence, uniqueness and continuous dependence on input data [15]. Motivated by Terzopoulos' pioneer work [14,4], we adapt his controlled-continuity surface model with linear constraints to solve a regularized and thus well-posed depth-recovery problem. The minimization to the resulting variational model with our specified linear constraints does not have, in general, an analytic solution. Thus it is necessary to discretize the continuous model such that high-speed computer solution is possible. The discretization can be done either in solution space (e.g., using the finite element methods) or in application data domain (e.g., using the finite difference methods). In this study we apply the latter technique to discretize the Terzopoulos' controlled-continuity model in [14] as a quadratic functional.

An extension of Terzopoulos' work is recently presented in [5] in which five types of point/curve features and their linear discretization on regular meshes are proposed. Since only the thin plate functional is used in [5] for smoothness constraint, as discussed in Section 3 of this paper, the surface model with thin-plate functional alone shows a property of strong bending stiffness, and thus, the so-called *fairing curve constraint* in [5] has to be applied to "soften" this surface stiffness. In this paper we also define and discretize a set of point/curve features in linear form. By applying the Lagrange multiplier method to minimize a quadratic functional with linear constraints, we can transform the general depth-recovery problem to a mathematically well defined *linearly constrained quadratic optimization problem*.

The linearly constrained quadratic optimization requires solving a large sparse linear system. In computer vision and computer graphics community, iterative numerical methods are widely used to solve large sparse

linear systems [16,17,5]. In this paper, by drawing observations from the special structures inherent in the linear system stemmed from our formulation of the depth-recovery problem, we show that a fast direct method with robustness and predictable behavior is possible.

Similar to us, Mallet [7] proposed a discrete smooth interpolation technique to model shapes over regular grids by minimizing quadratic functional with constrained features. Recently, this discrete interpolation method is applied to a novel texture mapping technique [6]. The major difference between Mallet's method and ours lies in that, for constraint modelling, Mallet directly removes redundant degrees of freedom from a quadratic minimization formulation, whereas we model the constraints with Lagrange multipliers. It is worth noting that since no boundary conditions are applied in Mallet's method, his formulation may possibly lead to a singular system. Further, the iterative method used in [7] depends both on a good initial guess and on the chosen iterative strategy. As a comparison, in our proposed technique, we devise a fast and efficient direct method for numerical computation.

1.2. Contributions of this paper

As a summary, the following contributions are made in this paper:

- To simulate and interpret the human perception, we propose a hierarchy of linear features for sketch-based free-form shape modelling. By using a finite difference discretization and applying the Lagrange multiplier method for the constraint modelling, we formulate and solve the general depth-recovery problem by minimizing a linearly constrained quadratic functional.
- An efficient direct method is proposed to solve a special linear system resulting from our formulation of the constrained quadratic optimization. By exploiting the special structures inherent in the linear system at hand, the proposed numerical engine is shown to be fast, stable and accurate.

2. Overview of the proposed method

The general depth-recovery problem is well known to be mathematically *ill-posed* [15,14]. Knowledge of visual perception can help *regularize* the model with stable solutions. Two general kinds of psychophysical object-detector models exist [3]. The first is a *regional binding* (or *grouping for similarity*) type of models that involves a comparison of any given texture (or luminance) element with its neighbors to find whether they are the same, so as to identify an area of relatively homogeneous texture

(or luminance). The second kind of models, known as *salient curves/corners detection* (or *boundary detection*), identify relatively large differences between neighboring elements by sharp spatial gradient.

In the proposed method, to simulate human perception with user intervention, in Section 3, we formulate the 2D regional binding information using a hybrid thin-plate surface model with controlled continuities, and in Section 4, we propose a hierarchy of linear features to characterize the 1D salient curve/corner information. In Section 5, we show that the depth values of sketch planes and thus the 3D free-form shape can be achieved by solving a linearly constrained quadratic optimization problem. An interesting observation is drawn from the special structures stemmed from the formulation of our specified optimization problem, from which we prove that system decomposition with dramatic reduction in computational cost is always possible. Noting that the special structure is only inherent in our formulation of the depth-recovery problem, this actually leads to a fast numerical engine with an interactive-rate performance that cannot be achieved by the standard numerical engines used in the most general case. To demonstrate the effectiveness and usefulness of the proposed method, two diverse applications are presented in Section 6. Finally our concluding remark is presented in Section 7.

3. A hybrid thin-plate surface model

In the proposed method, to characterize the regional binding information in the sketch plane with various continuity properties, we use a variation of the Terzopoulos' hybrid thin-plate model [4] given by

$$\mathcal{S}(z) = \frac{1}{2} \iint_{\Omega} \{ \tau(x, y)(z_{xx}^2 + 2z_{xy}^2 + z_{yy}^2) + [1 - \tau(x, y)](z_x^2 + z_y^2) \} dx dy, \quad (1)$$

where $z(x, y)$ is a surface function, $\Omega = (x, y) \subset \mathcal{R}^2$ describes an interested region in the sketch plane and $\tau(x, y)$ is a weighting function whose range is $[0, 1]$. Note that

- Minimization of $\lim_{\tau(x,y) \rightarrow 0} \mathcal{S}(z)$ locally characterizes a membrane spline, i.e., a C^0 surface (more accurately, a piecewise C^1 surface). Since the membrane is assumed to be so flexible as to give no resistance to bending, the work of deformation must be entirely owing to the change of the membrane's area. Thus under membrane energy, the surface is deformed sharply around the point feature constraints and is decayed with ripples (cf. Fig. 2b).
- Minimization of $\lim_{\tau(x,y) \rightarrow 1} \mathcal{S}(z)$ locally characterizes a thin-plate spline, i.e., a C^1 surface with continuous

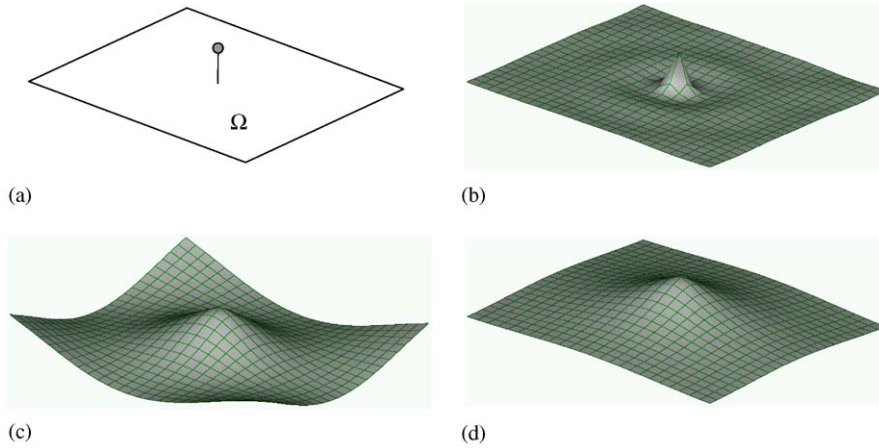


Fig. 2. Different shape characterization by adjusting the weighting parameter function $\tau(x, y)$. In (b) forward difference is applied for the first derivative approximation; see Section 4.1 for details of point interpolation: (a) An interpolating feature point; (b) Membrane surface modeling with $\tau = 0$; (c) Thin plate surface modeling with $\tau = 1$ and (d) Hybrid thin plate surface under tension with $\tau = 0.5$.

first derivative. Under thin-plate energy, surface force that acts at the boundary surface of the elastic body exhibits, and thus, gives rise to the strain potential energy that leads to strong bending effect on the resulting surface (cf. Fig. 2c). Note that if the surface map z is isometric, then the thin-plate energy is equal to the total curvature over the surface.

- Adjusting the weighting $\tau \in (0, 1)$ over the domain $\Omega \subset \mathbb{R}^2$ leads to a hybrid thin-plate surface under tension, which characterizes shape with the parameter function $\tau(x, y)$ (cf. Fig. 2d).
- Unlike the original Terzopoulos’ model in [4], the surface model characterized by Eq. (1) does not explicitly handle the discontinuity. In our application scenarios, we generally suppose that the discontinuity information is pre-specified before performing the numerical computation, i.e., discontinuity detection is not incorporated as part of the numerical computation. Then in the proposed method, discontinuities are constructed by geometrically trimming the surface with user-specified information (cf. Fig. 10).

As already alluded in Section 1.1, we are interested in discrete surface representation. The deduction of the continuous system depicted in Eq. (1) to an “equivalent” discrete (lumped parameter) system can be processed in several avenues, notably finite difference methods, finite element methods, boundary element methods and the method of lines. Since in our application scenarios the independent data domain Ω is in \mathbb{R}^2 , the finite difference method is adopted in the proposed approach; accordingly, the continuous domain $\Omega \subset \mathbb{R}^2$ is replaced by a quadrilateral mesh with fixed mesh-size.

Discretization of data domain. Let Ω be a bounded planar region with boundary Γ (cf. Fig. 3). We construct a discrete representation Ω^h of Ω as follows. Given an original point (x_0, y_0) and a mesh size $h > 0$, let \mathcal{R}_h^2 be the set of all points $(x_0 + ih, y_0 + jh)$, $i, j \in \mathbb{Z}$. We say that two points (x, y) and (x', y') of \mathcal{R}_h^2 are *adjacent* if $(x - x')^2 + (y - y')^2 = h^2$. Two adjacent points of \mathcal{R}_h^2 are *properly adjacent* if both are in $\Omega \cup \Gamma$ and if the open segment joining them (not necessarily including the endpoints) is in Ω . Let $\Omega^h = \Omega \cap \mathcal{R}_h^2$. A point p of Ω^h is *regular* if its four adjacent mesh points in \mathcal{R}_h^2 are properly adjacent to p . Let Γ^h denote all points of Ω^h which are not regular. Clearly Γ^h is a discrete approximation of Γ . Next we develop discrete representations of the differential operators in Eq. (1).

Discretization of thin-plate term. To discretize the thin-plate term in Eq. (1), Terzopoulos [4] and Zhang et al. [5] use the following formula to discretize second order partial derivatives at a regular node (i, j) with fixed mesh-size h in both directions:

$$\left. \frac{\partial^2 z}{\partial x^2} \right|_{(i,j)} = \frac{1}{h^2} (z_{i+1,j}^h - 2z_{i,j}^h + z_{i-1,j}^h) + O(h^2),$$

$$\left. \frac{\partial^2 z}{\partial y^2} \right|_{(i,j)} = \frac{1}{h^2} (z_{i,j+1}^h - 2z_{i,j}^h + z_{i,j-1}^h) + O(h^2),$$

$$\left. \frac{\partial^2 z}{\partial x \partial y} \right|_{(i,j)} = \frac{1}{h^2} (z_{i+1,j+1}^h - z_{i+1,j}^h - z_{i,j+1}^h + z_{i,j}^h) + O(h).$$

This gives rise to an asymmetric finite difference stencil with $O(h)$ truncation error (cf. Fig. 4a). In our practice, we use the following approximation for the

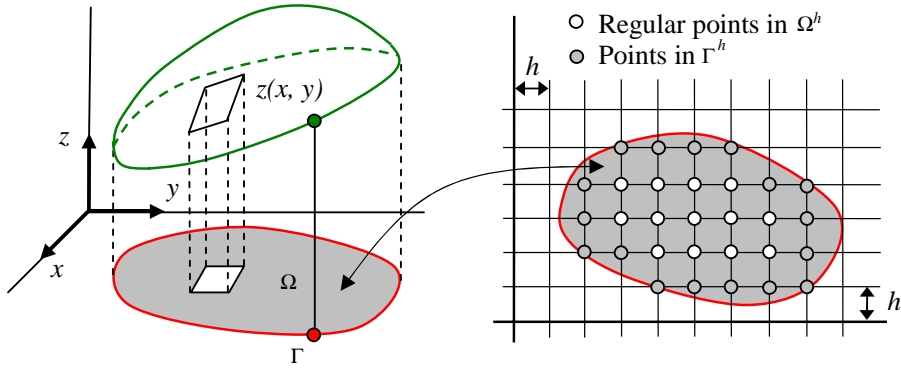


Fig. 3. Data domain discretization for the depth-recovery problem.

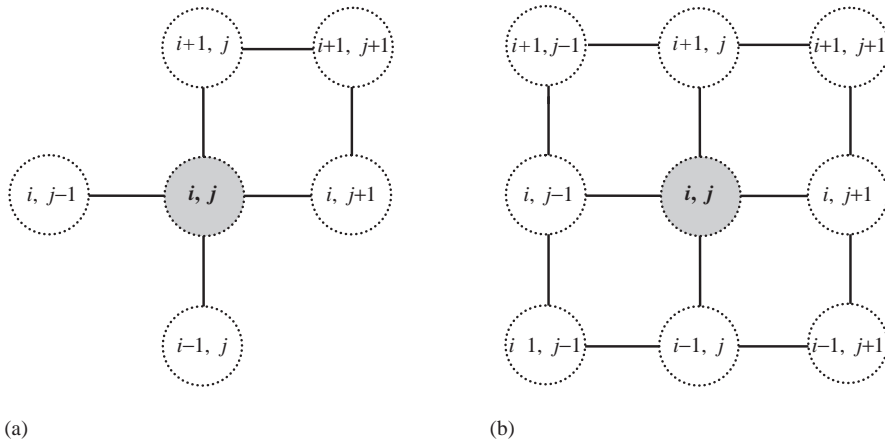


Fig. 4. The finite difference stencil at node (i, j) for thin-plate term discretization: (a) $O(h)$ approximation and (b) $O(h^2)$ approximation.

second-order mixed partial derivative:

$$\frac{\partial^2 z}{\partial x \partial y} \Big|_{(i,j)} = \frac{1}{4h^2} (z_{i+1,j+1}^h - z_{i+1,j-1}^h - z_{i-1,j+1}^h + z_{i-1,j-1}^h) + O(h^2),$$

which leads to a symmetric stencil with $O(h^2)$ truncation error (cf. Fig. 4b). Note that using the symmetric stencil can handle the nodes near irregular boundary much easier than using the asymmetric one.

Discretization of membrane term. We can discretize the membrane term in Eq. (1) using the following center difference to achieve the same order $O(h^2)$ of truncation error as that in thin-plate discretization:

$$\frac{\partial z}{\partial x} \Big|_{(i,j)} = \frac{1}{2h} (z_{i+1,j}^h - z_{i-1,j}^h) + O(h^2),$$

$$\frac{\partial z}{\partial y} \Big|_{(i,j)} = \frac{1}{2h} (z_{i,j+1}^h - z_{i,j-1}^h) + O(h^2).$$

While the above formula has a second-order truncation error, this does not mean that its utilization always offers a better approximation of the following forward

difference formula with the first-order truncation error $O(h)$:

$$\frac{\partial z}{\partial x} \Big|_{(i,j)} = \frac{1}{h} (z_{i+1,j}^h - z_{i,j}^h) + O(h),$$

$$\frac{\partial z}{\partial y} \Big|_{(i,j)} = \frac{1}{h} (z_{i,j+1}^h - z_{i,j}^h) + O(h).$$

Actually in our application scenario, the center difference always leads to an unstable numerical scheme; this is easy to interpret with the illustration in Fig. 5.

Substitution of both discretized thin-plate and membrane terms into Eq. (1) yields the following quadratic functional:

$$\begin{aligned} \mathcal{J}^h(\mathbf{z}^h) &= \frac{1}{16h^2} \sum_{(i,j) \in \Omega^h} \{ \tau_{ij}^h [8(z_{i+1,j}^h - 2z_{ij}^h + z_{i-1,j}^h)^2 \\ &\quad + (z_{i+1,j+1}^h - z_{i+1,j-1}^h - z_{i-1,j+1}^h + z_{i-1,j-1}^h)^2 \\ &\quad + 8(z_{i,j+1}^h - 2z_{ij}^h + z_{i,j-1}^h)^2] \\ &\quad + 8[1 - \tau_{ij}^h] [(z_{i+1,j}^h - z_{ij}^h)^2 + (z_{i,j+1}^h - z_{ij}^h)^2] \} \\ &= \frac{1}{2} \mathbf{z}^{hT} \mathbf{Q} \mathbf{z}^h, \end{aligned} \tag{2}$$

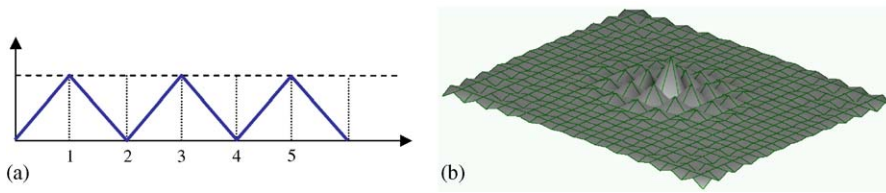


Fig. 5. Unstable numerical scheme by using center difference for the first derivative approximation: (a) A 1D shape which has zero first derivative at any node 1–5 when the center difference is applied and (b) Membrane surface modeling with $\tau = 0$ and with the center difference applied; compared to Fig. 2b.

where \mathbf{z}^h is a column vector containing depth values of the n mesh nodes of Ω^h , \mathbf{Q} is an $n \times n$ symmetric matrix and the factor $\frac{1}{2}$ is set for later convenience.

4. A hierarchy of linear sketch features

To model the 1D salient curves/corners information, we propose a hierarchy of linear² sketch features as illustrated in Fig. 6. In the proposed hierarchy, there are two fundamental linear geometric entities, points and normals. Other linear feature entities, such as Hermite data (point with normal) and characteristic curves, are regarded as the composition of the two fundamental entities. It is worth emphasizing that we set all sketch features to be linear with the due consideration of

- (1) linear interpolant features are intuitive for shape design/modification, and
- (2) in the numerical engine to be proposed in Section 5, linear features can guarantee the existence of a unique solution and make the numerical computation process more efficient.

4.1. Fundamental features

Point feature. A point feature specifies the depth value at a given point of $\Omega \subset \mathcal{R}^2$. Upon mesh discretization, the given point is snapped to its nearest mesh node in Ω^h . Every point feature $z^h(x_i, y_j) = d_{ij}$ offers one linear constraint on \mathbf{z}^h :

$$(\dots, 0, 1, 0, \dots)(\dots, z_{ij-1}^h, z_{ij}^h, z_{ij+1}^h, \dots)^T = d_{ij}.$$

One example of point feature interpolation is illustrated in Fig. 2.

Normal feature. Note that for a Monge surface $\mathcal{M}(x, y) = (x, y, z(x, y))$, its normal field takes the form of $(-\partial z/\partial x, -\partial z/\partial y, 1)$, subject to normalization. Then every normal feature $\mathbf{n}^h(x_i, y_j) = (n_x, n_y, n_z)$ offers

²Here *linear feature* means, upon discretization, the feature can be represented by vector forms; thus geometric position and normal are linear features, while curvature is not.

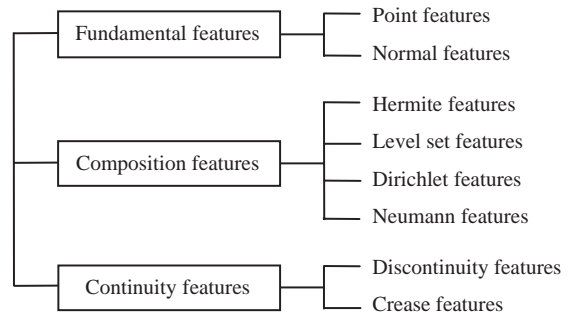


Fig. 6. The proposed hierarchy of linear sketch features.

two linear constraints:

$$\frac{\partial z}{\partial x} \Big|_{(i,j)} = \frac{1}{h} (z_{i+1,j}^h - z_{i,j}^h) = -\frac{n_x}{n_z},$$

$$\frac{\partial z}{\partial y} \Big|_{(i,j)} = \frac{1}{h} (z_{i,j+1}^h - z_{i,j}^h) = -\frac{n_y}{n_z}.$$

One example of normal feature interpolation is illustrated in Fig. 7.

4.2. Composition features

Hermite feature. The most direct composition feature is the *Hermite data*, i.e., a specified depth value together with a normal vector at a given mesh node. Each Hermite feature offers three linear constraints on the depth vector \mathbf{z}^h .

Level set feature. It is a composition feature grouping a set of point features. Given a curve specified over the domain $\Omega \subset \mathcal{R}^2$, we can convert it to a polygonal curve over Ω^h of which each node is assigned the same depth value; that is why it is called *level set curve*. This composition feature has a direct application to modeling iso-elevation contour maps. Two examples of shape modelling with two level set curves interpolation are illustrated in Figs. 8 and 9b.

Dirichlet feature. The Dirichlet problem is known as finding a function that is harmonic in Ω and takes the

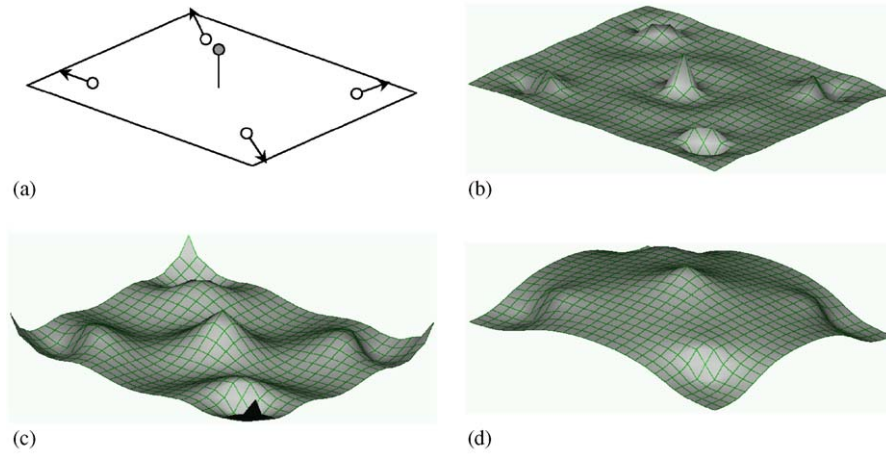


Fig. 7. Shape modelling by interpolating one point and four normal features: (a) One point and four normal features; (b) Shape modeling with $\tau = 0$; (c) Shape modeling with $\tau = 1$ and (d) Shape modeling with $\tau = 0.5$.

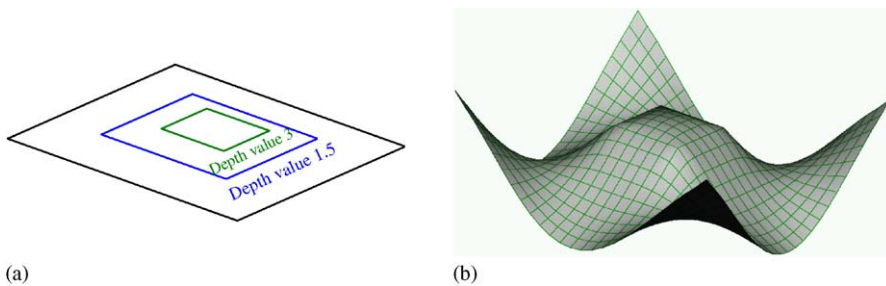


Fig. 8. Shape modelling by interpolating two level set curve features: (a) Two level set curve features and (b) Shape modeling with $\tau = 1$.

given values on the boundary Γ of Ω . In the proposed system we define the Dirichlet feature as a composition of point features to specify the boundary condition of the mesh Ω^h . One example of shape modelling with Dirichlet feature is illustrated in Fig. 9(c).

Neumann feature. The Neumann problem is known as finding a function that is harmonic in Ω and whose normal derivative takes the given values on the boundary Γ of Ω . In the proposed system we define the Neumann feature as a composition of the normal features satisfying

$$\frac{\partial z}{\partial \mathbf{n}} \Big|_{(i,j)} = \nabla z|_{(i,j)} \cdot \mathbf{n}|_{(i,j)}.$$

One example of shape modelling with Neumann feature is illustrated in Fig. 9(d).

In the proposed feature hierarchy, all the above features, including both fundamental and composition features, can be assembled into the following linear form:

$$\mathbf{C} \cdot \mathbf{z}^h = \mathbf{d},$$

where \mathbf{C} is an $m \times n$ constraint matrix with rank m ($m < n$) and \mathbf{d} is an m -vector.

4.3. Controlled-continuity features

In the proposed feature hierarchy, the rest features for shape characteristics concern with the geometric continuity. Benefited from the controlled-continuity surface model depicted in Eq. (1) and its discrete form depicted in Eq. (2), the features characterizing discontinuities (C^{-1}) and creases (C^0) on a hybrid thin-plate surface under tension (C^1) are readily obtained as follows.

Discontinuity feature. In the proposed system, the discontinuity information is assumed to be available prior to the numerical computation of the 3D shape. Given a curve \mathcal{D} on the domain Ω^h identifying a discontinuity on the underlying surface, we construct the discontinuity by duplicating the curve \mathcal{D} and forming a closed boundary consisting of \mathcal{D} and its duplicate \mathcal{D}' . One example of shape modelling with discontinuity features is illustrated in Fig. 10.

Crease feature. Crease features are modelled by first scan converting the curve \mathcal{C} that identifies the position

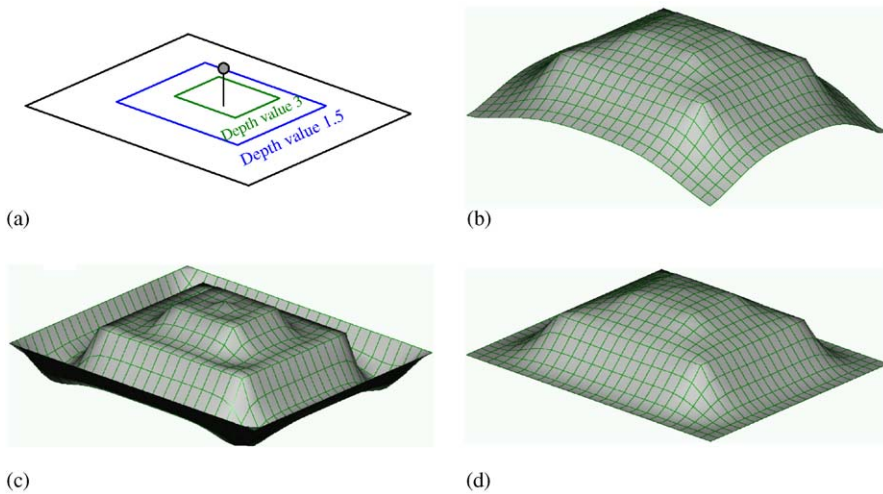


Fig. 9. Shape modelling with one point and two level set curve features and with $\tau = 0.5$: (a) One point and two level set curve features; (b) Shape modeling with free boundary condition; (c) Shape modeling with Dirichlet feature (all the boundary points are fixed at $z = 0$) and (d) Shape modeling with Neumann feature (normals at boundary are all parallel to z -axis).

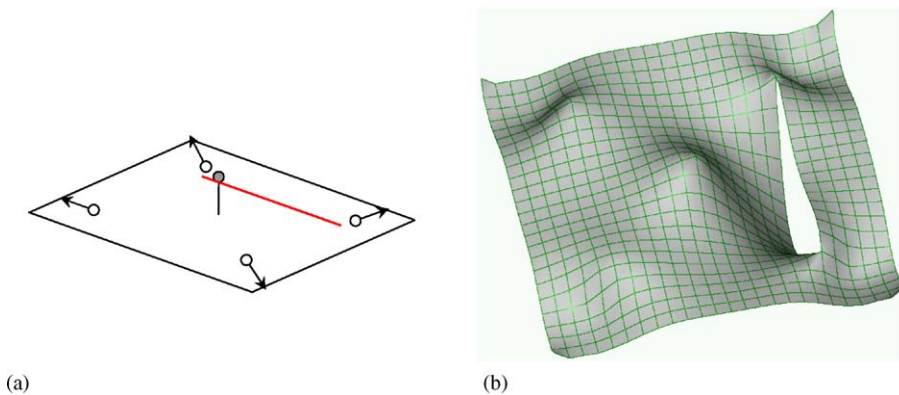


Fig. 10. Shape modelling with discontinuity feature: (a) One point, four normal and a discontinuity features and (b) The resulting mesh modeling with $\tau = 1$.

of creases and secondly dropping the smoothness term

$$\begin{aligned} &\tau_{i,j}^h [8(z_{i+1,j}^h - 2z_{i,j}^h + z_{i-1,j}^h)^2 + (z_{i+1,j+1}^h - z_{i+1,j-1}^h - z_{i-1,j+1}^h \\ &+ z_{i-1,j-1}^h)^2 + 8(z_{i,j+1}^h - 2z_{i,j}^h + z_{i,j-1}^h)^2] \\ &+ 8[1 - \tau_{i,j}^h][(z_{i+1,j}^h - z_{i,j}^h)^2 + (z_{i,j+1}^h - z_{i,j}^h)^2] \end{aligned}$$

of all nodes (i, j) lying on \mathcal{C} from the summation in Eq. (2). One example of shape modelling with crease features is shown in Fig. 11.

5. A fast and stable numerical engine

Recall that in the proposed system, 2D regional binding information is characterized by minimizing a

quadratic functional and 1D salient curve/corner information is characterized by a hierarchy of linear interpolatory features. Thus the sketch based 3D shape can be recovered by solving the following linear constrained quadratic minimization problem:

- * Find a solution //minimize a quadratic functional \mathcal{S}^h

$$\mathbf{u} = \arg \min_{\mathbf{z}^h \in \Omega^h} \{ \mathcal{S}^h(\mathbf{z}^h) \}$$

$$= \frac{1}{2} \mathbf{z}^{hT} \mathbf{Q} \mathbf{z}^h : \Omega^h \subset \mathcal{R}^n \rightarrow \mathcal{R},$$
- * Subject to $\mathbf{C} \mathbf{u} = \mathbf{d}$ //linear, interpolatory sketch feature constraints.

(3)

In Section 5.1 we show that \mathbf{Q} is a $n \times n$ matrix of rank n and \mathbf{C} is an $m \times n$ matrix of rank m , $m < n$. Thus

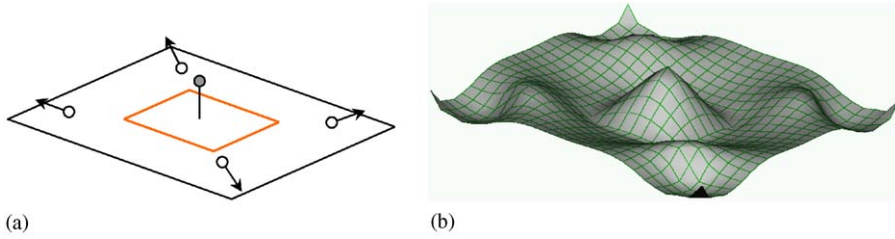


Fig. 11. Shape modelling with crease feature: (a) One point, four normal and a crease features and (b) The resulting mesh modeling with $\tau = 1$.

we can apply the Lagrange multiplier method to show that a necessary condition for the functional \mathcal{S}^h to have a relative extremum at the point $\mathbf{u} \in \mathcal{U}$ with respect to the set $\mathcal{U} = \{\mathbf{v}^h \in \Omega^h : \mathbf{C}\mathbf{v}^h = \mathbf{b}\}$ is the existence of a solution $(\mathbf{u}, \boldsymbol{\lambda}) \in \mathcal{R}^{n+m}$ of the linear system

$$\begin{cases} \mathbf{Q}\mathbf{u} + \mathbf{C}^T\boldsymbol{\lambda} = \mathbf{0} \\ \mathbf{C}\mathbf{u} = \mathbf{d} \end{cases} \implies \begin{pmatrix} \mathbf{Q} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{d} \end{pmatrix}. \quad (4)$$

Let

$$\mathbf{A}_{(n+m) \times (n+m)} = \begin{pmatrix} \mathbf{Q}_{n \times n} & (\mathbf{C}^T)_{n \times m} \\ \mathbf{C}_{m \times n} & \mathbf{0}_{m \times m} \end{pmatrix}.$$

Given that the $(n + m) \times (n + m)$ matrix \mathbf{A} is non-singular, a unique solution to the linear system (4) exists theoretically and is readily obtained by applying standard routines in a general numerical engine [18], e.g., a popular linear system solver—Gaussian elimination with partial pivoting (GEPP)—can be performed in $\frac{2}{3}(n + m)^3$ flops.³ However, it will be unwise if the sophisticated black-box numerical engine is used without the due considerations of the particular problem at hand. Indeed, exploiting special structures inherent in the problem can always lead to *faster* solutions, e.g., the solver with Cholesky factorization for a symmetric positive definite matrix \mathbf{A} can be performed in $\frac{1}{3}(n + m)^3$ flops, a factor of 2 better than the GEPP. Meanwhile the *stability* of an algorithm also depends on the problem: a well known example is the modified Gram–Schmidt method, which is stable when used to solve the least squares problem but can give poor results when used to compute an orthonormal basis of a matrix [19]. In the following subsections, we exploit in depth the special structures in our specified formulation (3) of the general depth-recovery problem and propose a fast and stable numerical engine for it.

5.1. Special structures in system (4)

First we draw observations from engineering literature that the linear system (4) is actually a special type of

³We normally state only the highest-order terms of flop counts. Thus for $\frac{2}{3}n^3$ flops, we really mean $\frac{2}{3}n^3 + O(n^2)$ flops.

equilibrium systems [20,21] which frequently rise in optimization, finite elements, structure analysis, and electrical networks. The following Lemma is in order.

Lemma 1. *The linear system (4) has the following structures:*

- (1) Matrix \mathbf{Q} is symmetric positive definite (SPD),
- (2) Matrix \mathbf{C} has rank m , i.e., full row rank,
- (3) Matrix \mathbf{A} is non-singular.

Proof.

- (1) First it is easy to see that \mathbf{Q} is symmetric; otherwise we can always transform it to a symmetric counterpart $\mathbf{Q}' = \frac{1}{2}(\mathbf{Q} + \mathbf{Q}^T)$ by

$$\begin{aligned} 2\mathcal{S}^h(\mathbf{z}^h) &= \mathbf{z}^{hT}\mathbf{Q}\mathbf{z}^h = (\mathbf{z}^{hT}\mathbf{Q}\mathbf{z}^h)^T \\ &= \mathbf{z}^{hT}\mathbf{Q}^T\mathbf{z}^h = \frac{1}{2}\mathbf{z}^{hT}(\mathbf{Q} + \mathbf{Q}^T)\mathbf{z}^h. \end{aligned}$$

Given an arbitrary vector \mathbf{z}^h , it is clear from Eq. (2) that $\mathcal{S}^h(\mathbf{z}^h) \geq 0$; the equality holds iff \mathbf{z}^h is in the form of $\mathbf{z}_0^h = a \cdot (1, 1, \dots, 1)^T$, where a is an arbitrary scalar, that physically means all mesh nodes have the same depth value and we call this state of the mesh surface as *lazy state* which is excluded from our proposed system. Thus $\mathcal{S}^h(\mathbf{z}^h) > 0$ for any arbitrary $\mathbf{z}^h \neq \mathbf{z}_0^h$ and the associated \mathbf{Q} is positive definite.

- (2) In the proposed system, by guaranteeing that each linear sketch feature specified by user is unique, the constraint matrix \mathbf{C} is of full row rank.
- (3) Let vectors $\mathbf{a} \in \mathcal{R}^n$ and $\mathbf{b} \in \mathcal{R}^m$ solve the homogeneous system $\mathbf{A}\begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} = \mathbf{0}$. Then $\mathbf{Q}\mathbf{a} + \mathbf{C}^T\mathbf{b} = \mathbf{0}$ and $\mathbf{C}\mathbf{a} = \mathbf{0} \implies \mathbf{a}^T\mathbf{Q}\mathbf{a} + \mathbf{a}^T\mathbf{C}^T\mathbf{b} = 0 \implies \mathbf{a}^T\mathbf{Q}\mathbf{a} = 0 \implies$ (since \mathbf{Q} is SPD) $\mathbf{a} = \mathbf{0} \implies \mathbf{C}^T\mathbf{b} = \mathbf{0} \implies \mathbf{b} = \mathbf{0}$. So only the null vector solves the homogeneous system with \mathbf{A} and therefore \mathbf{A} is non-singular. \square

The above lemma reveals the special structures in the coefficient matrix \mathbf{A} . When comparing our system (4) to the general form of equilibrium system [21], our system possesses a null vector in the upper right-hand side $\begin{pmatrix} \mathbf{0} \\ \mathbf{b} \end{pmatrix}$ of

the system (4). As revealed in the following theorem, more exciting structures exist in our proposed system.

Theorem 2. The solution $\begin{pmatrix} \mathbf{u} \\ \lambda \end{pmatrix}$ to the linear system (4), i.e.,

$$\begin{pmatrix} \mathbf{Q} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{d} \end{pmatrix},$$

is mathematically⁴ equal to the solution to the system

$$\begin{pmatrix} \mathbf{Q} & \mathbf{C}^T \\ \mathbf{0} & \mathbf{CQ}^{-1}\mathbf{C}^T \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{d} \end{pmatrix}, \quad (5)$$

i.e., the $(m+n) \times (m+n)$ linear system (4) can be solved with two smaller subsystems:

- (1) Calculate the m -vector λ by solving the $m \times m$ linear subsystem $-\mathbf{CQ}^{-1}\mathbf{C}^T\lambda = \mathbf{d}$;
- (2) Calculate the n -vector \mathbf{u} by solving the $n \times n$ linear subsystem $\mathbf{Q}\mathbf{u} = -\mathbf{C}^T\lambda$.

Proof. Since \mathbf{Q} is SPD, the Cholesky factorization $\mathbf{Q} = \mathbf{GG}^T$ exists. Then the LU factorization of the matrix \mathbf{A} is

$$\begin{aligned} \mathbf{A}_{(n+m) \times (n+m)} &= \begin{pmatrix} \mathbf{Q}_{n \times n} & (\mathbf{C}^T)_{n \times m} \\ \mathbf{C}_{m \times n} & \mathbf{0}_{m \times m} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{G}_{n \times n} & \mathbf{0}_{n \times m} \\ \mathbf{X}_{m \times n} & \mathbf{Y}_{m \times m} \end{pmatrix} \begin{pmatrix} (\mathbf{G}^T)_{n \times n} & (\mathbf{X}^T)_{n \times m} \\ \mathbf{0}_{m \times n} & (\mathbf{Z}^T)_{m \times m} \end{pmatrix} \end{aligned}$$

subject to

$$\mathbf{GX}^T = \mathbf{C}^T \quad \text{and} \quad \mathbf{XX}^T = -\mathbf{YZ}^T,$$

where both \mathbf{Y} and \mathbf{Z} are lower triangular matrices. Since \mathbf{XX}^T is SPD (cf. Corollary 3 below), \mathbf{Y} and \mathbf{Z} can be therefore uniquely determined by another Cholesky factorization $\mathbf{XX}^T = \mathbf{WW}^T$ with $\mathbf{Y} = -\mathbf{Z} = \mathbf{W}$. Thus we have

$$\begin{cases} \mathbf{GG}^T = \mathbf{Q} \\ \mathbf{GX}^T = \mathbf{C}^T \end{cases} \implies \mathbf{WW}^T = \mathbf{XX}^T = \mathbf{CQ}^{-1}\mathbf{C}^T$$

and

$$\mathbf{A} = \begin{pmatrix} \mathbf{Q} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{X} & \mathbf{W} \end{pmatrix} \begin{pmatrix} \mathbf{G}^T & \mathbf{X}^T \\ \mathbf{0} & -\mathbf{W}^T \end{pmatrix}.$$

Given the above LU factorization, the solution to the system (4) is to perform in tandem the forward substitution

$$\begin{pmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{X} & \mathbf{W} \end{pmatrix} \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{d} \end{pmatrix}$$

and the back substitution

$$\begin{pmatrix} \mathbf{G}^T & \mathbf{X}^T \\ \mathbf{0} & -\mathbf{W}^T \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix}.$$

Given the special null vector $\mathbf{0}$ in the right-hand side $\begin{pmatrix} \mathbf{0} \\ \mathbf{d} \end{pmatrix}$ of the system, the forward substitution gives

$$\mathbf{y}_1 = \mathbf{0} \quad \text{and} \quad \mathbf{W}\mathbf{y}_2 = \mathbf{d}$$

and the back substitution gives

$$\begin{aligned} \begin{pmatrix} \mathbf{G}^T & \mathbf{X}^T \\ \mathbf{0} & -\mathbf{W}^T \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \lambda \end{pmatrix} &= \begin{pmatrix} \mathbf{0} \\ \mathbf{y}_2 \end{pmatrix} \\ \implies \begin{cases} \mathbf{G}^T\mathbf{u} + \mathbf{X}^T\lambda = \mathbf{0} \implies \mathbf{GG}^T\mathbf{u} + \mathbf{GX}^T\lambda = \mathbf{0} \implies \mathbf{Q}\mathbf{u} = -\mathbf{C}^T\lambda \\ -\mathbf{W}^T\lambda = \mathbf{y}_2 \implies -\mathbf{WW}^T\lambda = \mathbf{W}\mathbf{y}_2 \implies -\mathbf{CQ}^{-1}\mathbf{C}^T\lambda = \mathbf{d}. \end{cases} \end{aligned}$$

That completes the proof. \square

We emphasize that both numbers m and n could be large in the proposed system, e.g., in the system leading to the example shown in Fig. 9d, $m = 227$ and $n = 625$. Therefore by Theorem 2 the system decomposition to the form (5) makes a big reduction in the computation cost.

Corollary 3. Given \mathbf{Q} is SPD and \mathbf{C} is of full row rank m , the matrix $\mathbf{CQ}^{-1}\mathbf{C}^T$ is also SPD.

Proof. Since \mathbf{Q} is SPD, the Cholesky factorization $\mathbf{Q} = \mathbf{GG}^T$ exists. Then, for an arbitrary vector \mathbf{x} , $\mathbf{x}^T\mathbf{CQ}^{-1}\mathbf{C}^T\mathbf{x} = \mathbf{x}^T\mathbf{C}\mathbf{G}^{-1}\mathbf{G}^{-1}\mathbf{C}^T\mathbf{x} = (\mathbf{G}^{-1}\mathbf{C}^T\mathbf{x})^T(\mathbf{G}^{-1}\mathbf{C}^T\mathbf{x}) \geq 0$, where the equality holds iff $\mathbf{x} = \mathbf{0}$ (since \mathbf{C} is of full rank m and \mathbf{G} is of full rank n). \square

Remark 4. It cannot be concluded (by simple comparison between the system (4) and the system (5)) that the matrix \mathbf{A} is reducible, i.e., there does not exist a permutation matrix \mathbf{P} such that

$$\mathbf{P}^{-1}\mathbf{A}\mathbf{P} = \begin{pmatrix} \mathbf{Q} & \mathbf{C}^T \\ \mathbf{0} & \mathbf{CQ}^{-1}\mathbf{C}^T \end{pmatrix}.$$

In our application scenario, \mathbf{A} is generally irreducible and the reason that the system decomposition (from system (4) to (5)) is possible is all due to the existence of a null sub-vector in the right-hand side $\begin{pmatrix} \mathbf{0} \\ \mathbf{d} \end{pmatrix}$ of the system (4).

Remark 5. Given any scalar $a \neq 0$ and $\mathbf{Q}' = a\mathbf{Q}$, the solution \mathbf{u}' to the system with \mathbf{Q}' is identical with the solution \mathbf{u} to the system with \mathbf{Q} . This means that different weighting between the quadratic functional and linear feature constraints provide the same result.

⁴The meaning of *mathematically* but not *physically* is presented in Remark 4 below.

Proof.

$$\begin{aligned} \begin{pmatrix} \mathbf{Q} & \mathbf{C}^T \\ \mathbf{0} & \mathbf{CQ}^{-1}\mathbf{C}^T \end{pmatrix} \begin{pmatrix} \mathbf{u}' \\ \lambda'/a \end{pmatrix} &= \begin{pmatrix} \mathbf{0} \\ \mathbf{d} \end{pmatrix} \\ \Rightarrow \begin{pmatrix} \mathbf{Q} & \mathbf{C}^T \\ \mathbf{0} & \mathbf{CQ}^{-1}\mathbf{C}^T \end{pmatrix} \begin{pmatrix} \mathbf{u}' \\ \frac{1}{a}\lambda' \end{pmatrix} & \\ = \begin{pmatrix} \mathbf{0} \\ \mathbf{d} \end{pmatrix}. & \quad \square \end{aligned}$$

5.2. Stability analysis

The conditional number of a linear system $\mathbf{Ax} = \mathbf{b}$ measures the sensitivity of the solution \mathbf{x} to perturbations in the input (\mathbf{A}, \mathbf{b}) . A system is called *ill-conditioned* if its conditional number is large. An important characteristic observed in the proposed system is that although by Lemma 1 the matrix \mathbf{A} is non-singular and the linear system (4) has a unique solution theoretically, the system (4) as well as the system (5) is frequently numerically rank-deficient and thus ill-conditioned.

The most crucial problem in connection with rank-deficient systems is how to reliably compute the numerical rank of the matrix \mathbf{A} . The canonical tool for numerical rank analysis is the singular value decomposition (SVD). Let $\mathbf{A} \in \mathbb{R}^{m \times n}$. Then the SVD of \mathbf{A} is

$$\begin{aligned} \mathbf{A} &= \mathbf{U}\mathbf{D}\mathbf{V}^T = [\mathbf{u}_1, \dots, \mathbf{u}_n] \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix} [\mathbf{v}_1, \dots, \mathbf{v}_n]^T \\ &= \sum_{i=1}^n \mathbf{u}_i \sigma_i \mathbf{v}_i^T, \end{aligned}$$

where $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$ are the singular values of \mathbf{A} and the vectors \mathbf{u}_i and \mathbf{v}_i are the i th left and right singular vector of \mathbf{A} , respectively. The 2-norm condition

number of \mathbf{A} is by definition

$$\text{cond}_2(\mathbf{A}) = \frac{\sigma_1}{\sigma_n}.$$

From the relation $\mathbf{A}\mathbf{v}_i = \sigma_i\mathbf{u}_i \Rightarrow \|\mathbf{A}\mathbf{v}_i\|_2 = \sigma_i$, it is clear that for a small singular value σ_i relative to $\sigma_1 = \|\mathbf{A}\|_2$, there exists a linear combination of the columns of \mathbf{A} which amount closely to a null vector, implying that \mathbf{A} is nearly rank deficient.

We should be more careful when investigating what is the source of ill conditioning of the proposed systems (4) and (5). If the ill-conditioned system is derived from a direct discretization of an ill-posed problem, all singular values of the coefficient matrix should decay gradually to zero, with no distinct gap anywhere in the spectrum; this situation is quite complicated since we cannot merely filter out a cluster of small singular values. Fortunately this is not our case. In our application scenario, we discretize a *regularized* ill-posed problem. Thus we are confronted with a *rank-deficient* problem but not a *discrete ill-posed* problem. The numerical deficiency comes from the constraint matrix \mathbf{C} : in the proposed interactive system, the user-specified features are usually non-uniformly distributed and then the constraint matrix \mathbf{C} in system (4) is always numerical deficient, so is $\mathbf{CQ}^{-1}\mathbf{C}^T$ in system (5). This conclusion is confirmed in our experiments since there is a well-determined gap between the large and small singular values of $\mathbf{CQ}^{-1}\mathbf{C}^T$; the singular value spectrums of two instantiations of the system (5) are illustrated in Fig. 12.

Given a well-determined gap between singular values σ_{r_ϵ} and $\sigma_{r_\epsilon+1}$ of \mathbf{A} , it makes sense to say that \mathbf{A} has numerical ϵ -rank r_ϵ , $\sigma_{r_\epsilon} > \epsilon \geq \sigma_{r_\epsilon+1}$. For identification of numerical rank deficiency, the predefined constant ϵ should be small enough when compared with the machine precision (e.g., 10^{-6} for single precision and 10^{-12} for double). As a consequence, a stabilized solution to the rank-deficient system $\mathbf{Ax} = \mathbf{b}$ is to simply ignore the SVD components associated with the small singular values and compute the truncated SVD (TSVD)

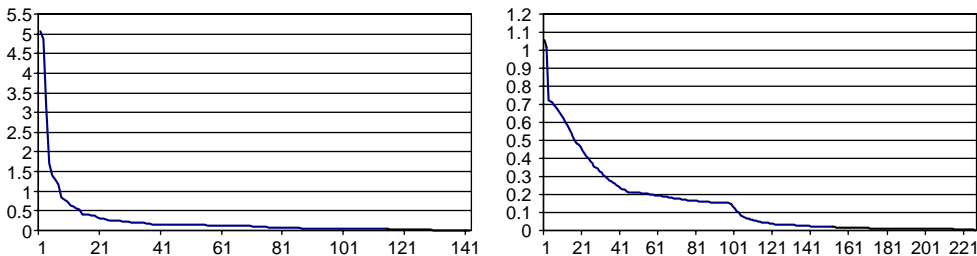


Fig. 12. The singular value spectrums of two instantiations of the system (5). Left: the plot of singular values of the matrix $\mathbf{C}^T\mathbf{Q}^{-1}\mathbf{C}$ with the size of 142×142 solved for the example in Fig. 9c. Right: the plot of the 227×227 matrix $\mathbf{C}^T\mathbf{Q}^{-1}\mathbf{C}$ solved for the example in Fig. 9d.

solution by

$$\mathbf{x} = \sum_{i=1}^{r_e} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i.$$

5.3. A fast truncated rank-revealing decomposition

We use the SVD for stability analysis of our proposed system. The SVD is a superior numerical tool that can reveal all difficulties in an ill-conditioned system. However, the cost paid for this “superiority” is a high computational cost, e.g., the Golub–Kahan–Reinsch SVD algorithm for a dense $m \times n$ matrix has the complexity of $14mn^2 + 8n^3$ flops. In practical applications, many other rank-revealing decompositions, such as RRQR and UTV decompositions, have been proposed as computationally advantageous alternatives to the SVD when the numerical rank is either high or low. In our practice, we implement a symmetric rank-revealing decomposition [22] as the numerical engine in the proposed system to achieve a truncated VSV solution [23] which we briefly summarize as follows with the emphasis on the special structures in our proposed system.

Let $\mathbf{M} = \mathbf{CQ}^{-1}\mathbf{C}^T$. In the proposed system (5), \mathbf{M} (as well as \mathbf{A} in the system (4)) is symmetric. We thus expect the matrix decomposition to inherit the symmetry. The symmetric counterpart of SVD is the eigenvalue decomposition (EVD) [24] and for the UTV decomposition, the symmetric counterpart is the VSV decomposition [22]:

$$\mathbf{M} = \mathbf{VSV}^T = (\mathbf{V}_1 \ \mathbf{V}_2) \begin{pmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{12}^T & \mathbf{S}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{pmatrix}, \quad (6)$$

where \mathbf{V} has orthonormal columns and \mathbf{S} is a symmetric matrix with \mathbf{S}_{11} being $r_e \times r_e$. The VSV decomposition is rank-revealing since the matrix \mathbf{S} is arranged in such a way that \mathbf{S}_{11} is well-conditioned ($\approx \sigma_1/\sigma_{r_e}$) and $\sqrt{\|\mathbf{S}_{12}\|_2^2 + \|\mathbf{S}_{22}\|_2^2}$ be of the order σ_{r_e+1} ($\approx \sqrt{\sigma_{r_e+1}^2 + \dots + \sigma_m^2}$). Given the VSV decomposition (6) of \mathbf{M} , the columns of \mathbf{V}_1 span an approximation to the numerical range $\mathcal{R}_{r_e}(\mathbf{M})$ of \mathbf{M} and the columns of \mathbf{V}_2 span an approximation to the numerical null space $\mathcal{N}_{r_e}(\mathbf{M})$ of \mathbf{M} . Therefore a stabilized truncated VSV solution to the ill-conditioned system $\mathbf{M}\boldsymbol{\lambda} = \mathbf{d}$ is

$$\boldsymbol{\lambda}_S = \mathbf{V}_1 \mathbf{S}_{11}^{-1} \mathbf{V}_1^T \mathbf{d}.$$

To perform the VSV decomposition, the algorithm consists of an initial triangular factorization and a rank-revealing postprocessing step. Since by Corollary 3 (resp., Lemma 1) the matrix \mathbf{M} (resp., \mathbf{A}) is SPD, the initial factorization can be achieved in $\frac{1}{3}m^3$ (resp., $\frac{1}{3}(m+n)^3$) flops by the symmetrically pivoted Cholesky

factorization $\mathbf{P}^T \mathbf{M} \mathbf{P} = \mathbf{G} \mathbf{G}^T$, \mathbf{P} is a permutation matrix. Due to $\sigma_i(\mathbf{G}) = \sqrt{\sigma_i(\mathbf{M})}$ (see Theorem 4.1 in [22]), the postprocessing step can be achieved by computing the ULV decomposition of $\mathbf{E}_X \mathbf{G}^T \mathbf{E}_X$, where \mathbf{E}_X is an exchange matrix consisting of the columns of the identity matrix in reverse order. The ULV decomposition requires only $c(m - r_e)m^2$ flops if $r_e \approx m$, or $cr_e m^2$ flops if $r_e \ll m$, where c is an algorithm-dependent constant.

5.4. Robustness in the presence of rounding errors

In the ideal situation, the above presented VSV solution to our proposed system (5) obviously works well. In real-world scenarios, numerical computation with finite precision arithmetic represents real numbers by approximate quantities. In this stage the leading source of errors is *rounding* that replaces a given real number by the nearest p significant digit number, p is a machine-dependent constant. It is inviting to ask what is the performance of the adopted truncated VSV solution in the presence of rounding errors with the following arguments.

By adopting a truncated VSV solution, we actually apply a *direct method* to solve the linear system (5). The use of *iterative methods*, however, has received intensive attentions in CV/CG applications [16,12,17,5]. A typical iterative method involves (a) the selection of an initial approximation $\boldsymbol{\lambda}_0$ to the solution $\boldsymbol{\lambda}$ of the linear system $\mathbf{M}\boldsymbol{\lambda} = \mathbf{d}$ and (b) the determination of a sequence $\{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \dots\}$ by performing some iterative algorithm, which, if the method is properly designed, will converge to the exact solution $\boldsymbol{\lambda}$. For e.g., in the context of fluid dynamics, a well-known iteration method for solving the equilibrium system (4) is Uzawa’s method, which resembles a relaxed block SOR iteration and the iteration process converges if the relaxation parameter satisfies $0 < \omega < 2/\lambda_{\max}(\mathbf{M})$. The use of iterative methods has the principal advantage that the matrix \mathbf{M} is not altered during the computation process; hence though the iteration process may be long, the problem of the accumulation of rounding errors is less serious than most direct methods, where the matrix is changed during the computation process. On the other hand, recent advances in matrix analysis [20,25,26] advocate using direct methods for equilibrium systems because of their predictable behavior. In our application scenario, if our adopted direct method is insensitive to the rounding errors, the underlying numerical engine inherent in the proposed system is certainly superior to the existing numerical engines with the core of iterative methods.

The effect of rounding errors can be illustrated by the mixed forward–backward error analysis. Consider the linear subsystem $\mathbf{M}\boldsymbol{\lambda} = \mathbf{d}$ in (5). Let $\tilde{\mathbf{M}} = \mathbf{M} + \mathbf{E}$, $\tilde{\mathbf{d}} = \mathbf{d} + \mathbf{e}$, $\mathbf{M} = \mathbf{VSV}^T$, $\tilde{\mathbf{M}} = \tilde{\mathbf{V}}\tilde{\mathbf{S}}\tilde{\mathbf{V}}^T$, where \mathbf{M} is given

exactly and \mathbf{E} represents the influence of rounding errors with unit roundoff \mathbf{e} whose norm is typically of order 10^{-8} or 10^{-16} in single and double precision computer arithmetic, respectively. For the statistical distribution of rounding errors in float point arithmetic, the reader is referred to [27] and the references therein.

Let $\lambda_S = \mathbf{V}_1 \mathbf{S}_{11}^{-1} \mathbf{V}_1^T \mathbf{d}$ and $\tilde{\lambda}_S = \tilde{\mathbf{V}}_1 \tilde{\mathbf{S}}_{11}^{-1} \tilde{\mathbf{V}}_1^T \tilde{\mathbf{d}}$. It can be shown (see Theorem 3.2 in [23]) that given $\|\mathbf{E}\|_2 \leq \sigma_{r_e+1} < \sigma_{r_e} - \|\mathbf{S}_{22}\|_2$,

$$\begin{aligned} & \frac{\|\lambda_S - \tilde{\lambda}_S\|_2}{\|\lambda_S\|_2} \\ & \leq \frac{\sigma_1}{\sigma_{r_e}} \left(\frac{\|\mathbf{e}\|_2}{\|\mathbf{M}\lambda_S\|_2} + \frac{\|\mathbf{E}\|_2}{\|\mathbf{M}\|_2} + \frac{\|\tilde{\mathbf{S}}_{12}\|_2}{\|\mathbf{M}\|_2} \right. \\ & \quad \left. + \sin \phi_{r_e} \frac{\|\mathbf{d} - \mathbf{M}\lambda_S\|_2}{\|\mathbf{M}\lambda_S\|_2} \right) + \sin \phi_{r_e}, \end{aligned}$$

where the distance $\sin \phi_{r_e}$ between the subspaces $\mathcal{R}_{r_e}(\mathbf{M})$ and $\mathcal{R}_{r_e}(\tilde{\mathbf{M}})$ is bounded by

$$\sin \phi_{r_e} \leq \frac{\|\mathbf{S}_{12}\|_2 + \|\tilde{\mathbf{S}}_{12}\|_2 + \|\mathbf{E}\|_2}{\sigma_{r_e} - \|\tilde{\mathbf{S}}_{22}\|_2}.$$

In our application scenario, since there is a distinct gap between σ_{r_e} and σ_{r_e+1} comparable to the roundoff unit, we can expect that $\|\mathbf{E}\|_2$, $\|\mathbf{e}\|_2$, $\|\mathbf{S}_{12}\|_2$, $\|\tilde{\mathbf{S}}_{12}\|_2$, $\|\tilde{\mathbf{S}}_{22}\|_2$ have the same order of machine precision $\sigma_{r_e+1} \approx \|\mathbf{e}\|_2$ and thus

$$\sin \phi_{r_e} \approx \frac{\|\mathbf{e}\|_2}{\sigma_{r_e}} \quad \text{and} \quad \frac{\|\lambda_S - \tilde{\lambda}_S\|_2}{\|\lambda_S\|_2} \approx \frac{\|\mathbf{e}\|_2}{\sigma_{r_e}} \left(c + \frac{\|\mathbf{M}\|_2}{\|\mathbf{M}\lambda_S\|_2} \right),$$

where c is a constant. That means in the presence of rounding errors, the error of the truncated VSV solution $\tilde{\lambda}_S$ relative to λ_S is bounded in the same order of machine precision. We thus conclude that our present numerical scheme is stable in the sense of rounding error insensitivity.

We finally note that we diagnose the ill-conditioning of the system using the SVD tool and in practice solve the ill-conditioned system using a truncated VSV solution due to its computational efficiency. It is also interesting to compare the truncate SVD and VSV solutions side by side in terms of accuracy. Let the EVD of \mathbf{M} be $\mathbf{U}\mathbf{D}^{-1}\mathbf{U}^T$ and the TSVD solution to $\mathbf{M}\lambda = \mathbf{d}$ be $\lambda_D = \mathbf{U}_1 \mathbf{D}_1^{-1} \mathbf{U}_1^T \mathbf{d}$, where \mathbf{U}_1 consists of the first r_e columns of \mathbf{U} . By Theorem 2.2 in [23], the truncated VSV and SVD solutions λ_S and λ_D satisfy

$$\frac{\|\lambda_D - \lambda_S\|_2}{\|\lambda_D\|_2} \leq \sin \theta_{r_e} \left[1 + \frac{\sigma_1}{\sigma_{r_e}} \left(\frac{\|\mathbf{S}_{12}\|_2}{\|\mathbf{M}\|_2} + \frac{\|\mathbf{d} - \mathbf{M}\lambda_D\|_2}{\|\mathbf{M}\lambda_D\|_2} \right) \right],$$

where in our case, the distance $\sin \theta_{r_e}$ between the subspaces spanned by columns of \mathbf{U}_1 and \mathbf{V}_1 , respectively, is bounded by $\sin \theta_{r_e} \approx (\sigma_{r_e+1})/\sigma_{r_e}$. Thus in our

application scenario,

$$\frac{\|\lambda_D - \lambda_S\|_2}{\|\lambda_D\|_2} \approx \frac{\|\mathbf{e}\|_2}{\sigma_{r_e}} + O(\|\mathbf{e}\|_2^2),$$

i.e., the difference between λ_S and λ_D is also in the order of machine precision.

6. Implementation and applications

We have implemented a prototype system for sketch-based free-form shape modelling with the fast and stable numerical engine proposed in this paper. The user interface of the system is illustrated in Fig. 13. Two applications are presented below to demonstrate the usefulness and effectiveness of the proposed system.

6.1. Car body design

In automobile industry, computer models designed by artists are preferred in advance of the physical prototypes. To design a computer model with sketching, the user creates a set of essential curves. The quality and aesthetics of these curves, so called *character lines*, determine the global product impression. In the proposed system, we first specify controlled-continuity features, e.g., the weighting parameter $\tau(x, y)$, creases and discontinuity features, such that the discrete quadratic \mathbf{Q} is fixed (cf. Fig. 14a) and then \mathbf{Q}^{-1} can be preprocessed. Subsequently, other aesthetic features are added without changing \mathbf{Q}^{-1} and thus, we can achieve interactive-rate editing of the 3D car body; one example is illustrated in Fig. 14b.

6.2. Cartoon character design

In entertainment industry, artists usually create cartoon pictures in 2D sketch planes. To generate the 3D effect of the 2D sketch-based cartoon character, the proposed system can be applied. One example of 3D effects on a cartoon character depicted in a single 2D sketch is illustrated in Fig. 15. Note that a single sketch may only offer a portion of the complete 3D shape. If several consistent character sketches from different viewpoints are provided, it is possible to build a complete 3D character by applying the standard range image registration, blending and stitching techniques. Note also that the stereopsis techniques cannot capture shapes accurately from a wide range of viewpoints and silhouette-based techniques cannot capture shapes with salient open curves and corners. It is also illustrative from Figs. 1 and 15 that the proposed system can be used for image-based modelling and rendering [8].

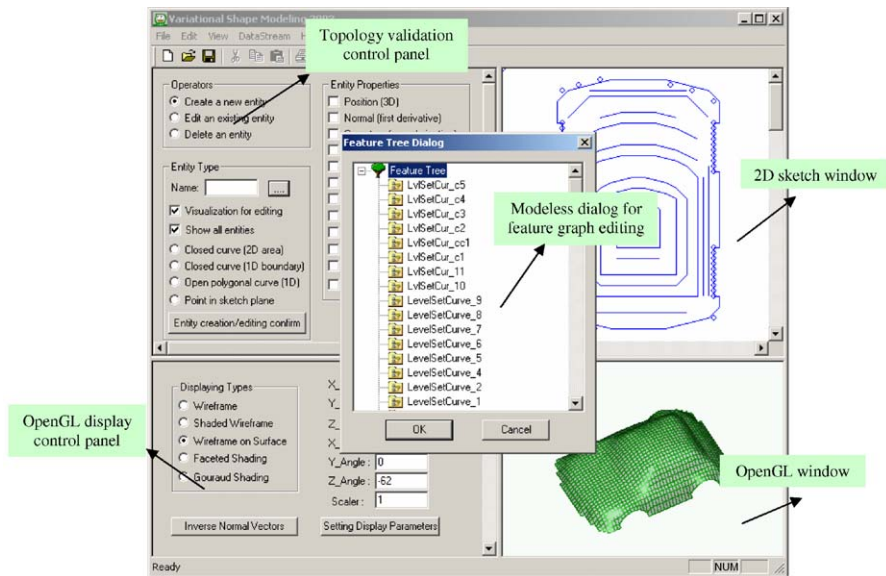


Fig. 13. The user interface of the prototype system for sketch-based shape design.

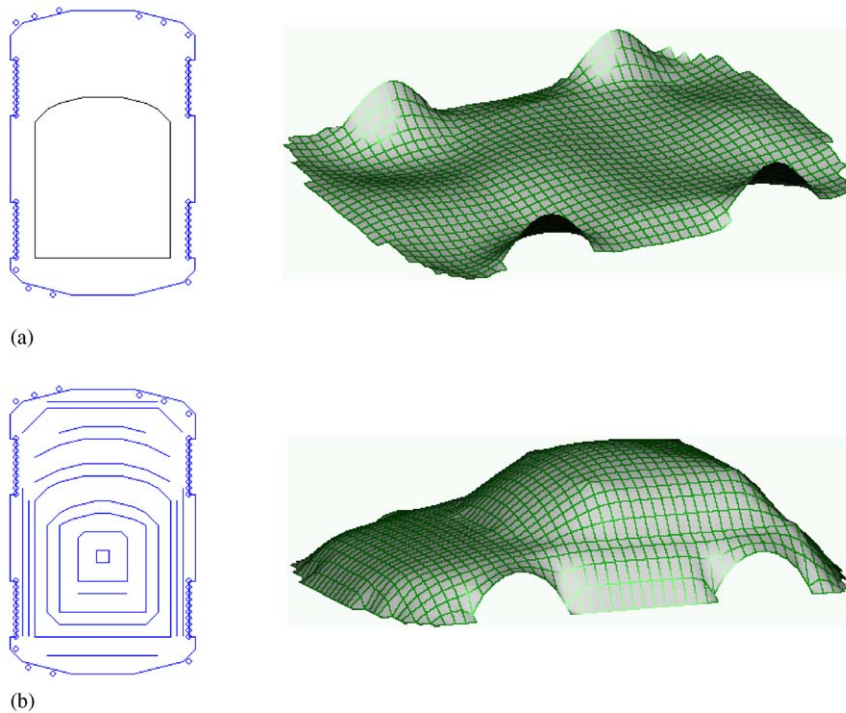


Fig. 14. The car body design process: (a) Specification of controlled-continuity features, together with a mixed boundary condition and (b) Car body design by adding more aesthetic features.

7. Summary

In this paper, a sketch-based free-form shape modeling technique is proposed by solving a linearly

constrained quadratic optimization problem with a fast and stable numerical engine. To infer the 3D shape information from 2D sketches, both clues of 2D region-binding and 1D salient curves/corners are exploited and

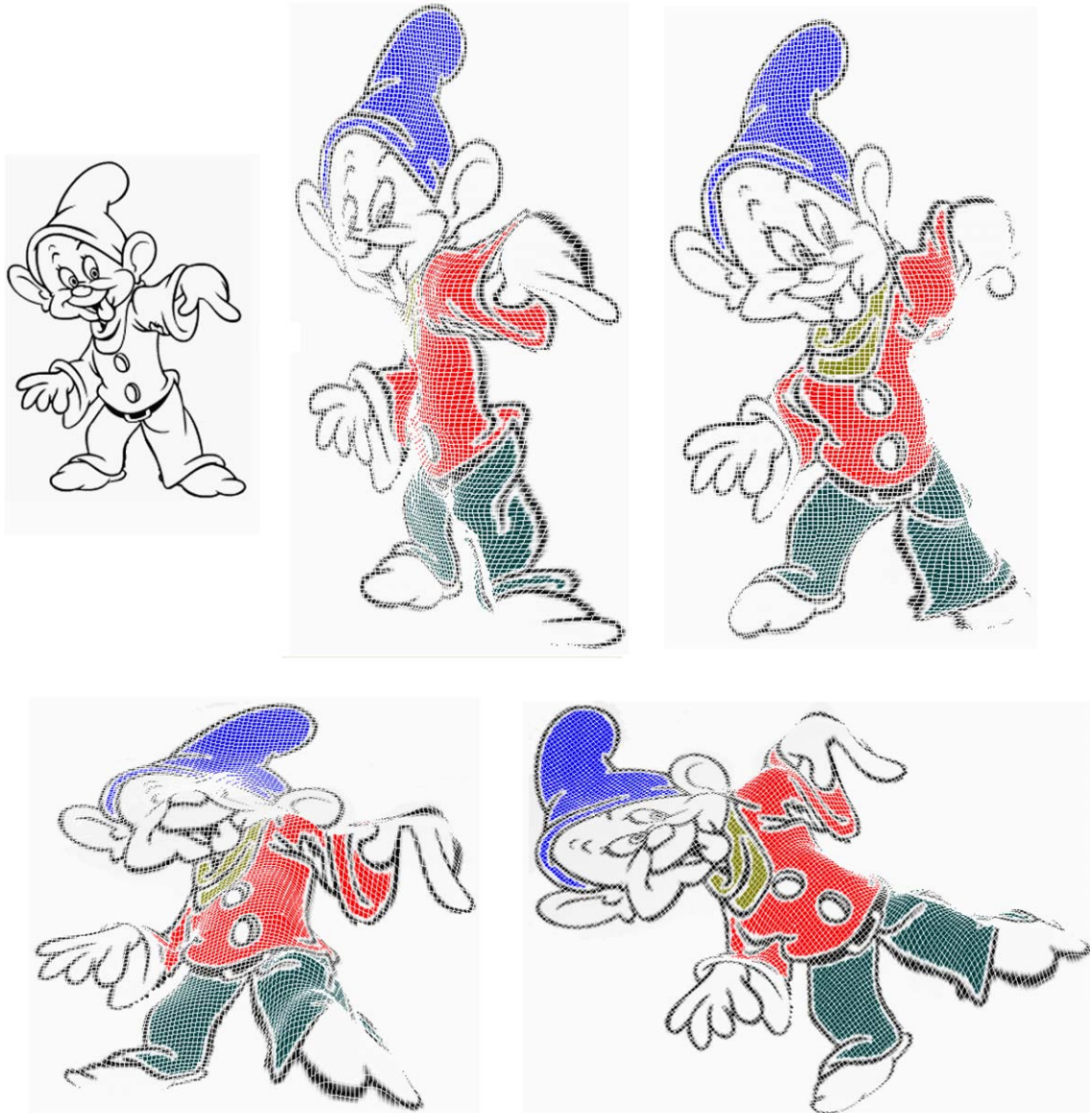


Fig. 15. Cartoon character design from sketches. The upper-left BW 2D cartoon image is downloaded from <http://www.coloring.ws/cartoons.htm>.

formulated by a hybrid thin-plate surface model and a hierarchy of linear features, respectively. To gain an interactive-rate performance in the proposed system, a fast and stable numerical engine is developed with a rigorous proof that system decomposition with reduced computational cost is always possible by exploiting the special structures inherent in the proposed system. Stability, accuracy and efficiency are studied in-depth for the proposed numerical engine. The usefulness and effectiveness of the proposed technique is finally demonstrated by two diverse applications with examples.

References

- [1] Metaxas D, Miliou E. Reconstruction of a color image from nonuniformly distributed sparse and noisy data. *Graphical Models and Image Processing* 1992;54(2):103–11.
- [2] Shen J. Inpainting and the fundamental problem of image processing. *SIAM News* 2003;36(5).
- [3] Sonka M, Hlavac V, Boyle R. *Image processing, analysis, and machine vision*, 2nd ed. Pacific Grove, CA: PWS Pub.; 1999.
- [4] Terzopoulos D. The computation of visible surface representations. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 1988;10(4):417–38.

- [5] Zhang L, Dugas-Phocion G, Samson J, Seitz S. Single-view modeling of free-form scenes. In: IEEE Proceedings of CVPR, vol. 1; 2001. 1-990–7.
- [6] Levy B, Mallet JL. Non-distorted texture mapping for sheared triangulated meshes. In: Proceedings of SIGGRAPH' 98, ACM SIGGRAPH; 1998. p. 19–24.
- [7] Mallet JL. Discrete smooth interpolation. ACM Transaction on Graphics 1989;8(2):121–44.
- [8] Ruprecht D, Muller H. Image warping with scattered data interpolation. IEEE Computer Graphics and Applications 1995;15(2):37–43.
- [9] Vazquez P, Marco J, Sbert M. Point-based modeling from a single image, Lecture Notes in Computer Science, vol. 3039, 2004. p. 245–51.
- [10] Nielson G. Scattered data modeling. IEEE Computer Graphics & Applications 1993;13(1):60–70.
- [11] Franke R, Nielson G. Scattered data interpolation and applications: a tutorial and survey. In: Hagen H, Roller D, editors. Geometric modeling: methods and their application. Berlin; New York: Springer-Verlag; 1991. p 131–60.
- [12] Schroder P, Sweldens W. Digital geometry process, SIGGRAPH Course Notes. Los Angeles, California, USA; 2001.
- [13] Zaharia M, Dorst L. Modeling and visualization of 3D polygonal mesh surfaces using geometric algebra. Computers & Graphics 2004;28(4):519–26.
- [14] Terzopoulos D. Regularization of inverse visual problems involving discontinuities. IEEE Transactions on Pattern Analysis & Machine Intelligence 1986;8(4):413–24.
- [15] Hansen P. Rank-deficient and discrete ill-posed problem. Philadelphia: SIAM; 1998.
- [16] Baraff D, Witkin A. Large step in cloth simulation. In: Proceedings of SIGGRAPH' 98, ACM SIGGRAPH; 1998. p. 43–54.
- [17] Terzopoulos D. Multiresolution computation of visible surface representations. PhD dissertation, Department of Electrical Engineering & Computer Sciences, MIT; 1984.
- [18] Press W, Teukolsky S, Vetterling W, Flannery B. Numerical recipes in C++: the art of scientific computing, 2nd ed. Cambridge: Cambridge University Press; 2002.
- [19] Golub G, van Loan C. Matrix computations, 3rd ed. Baltimore, Md.: Johns Hopkins University Press; 1996.
- [20] Liu YJ, Tang K, Yuen MMF. Efficient and stable numerical algorithms on equilibrium equations for geometric modeling. In: Geometric Modeling and Process; 2004. p. 291–300.
- [21] Strang G. A framework for equilibrium equations. SIAM Review 1988;30(2):283–97.
- [22] Hansen P, Yalamov P. Computing symmetric rank-revealing decompositions via triangular factorization. SIAM Journal on Matrix Analysis & Applications 2001; 23(2):443–58.
- [23] Fierro R, Hansen P. Truncated VSV solutions to symmetric rank-deficient problems. Bits 2002;42(3): 531–40.
- [24] Bunse-Gerstner A, Gragg W. Singular value decompositions of complex symmetric matrices. Journal of Computational and Applied Mathematics 1988;21(1):41–54.
- [25] Tuma M. A note on the LDL^T decomposition of matrices from saddle-point problems. SIAM Journal on Matrix Analysis & Applications 2002;23(4):903–15.
- [26] Vavasis S. Stable numerical algorithms for equilibrium systems. SIAM Journal on Matrix Analysis & Applications 1994;15(4):1108–31.
- [27] Barlow J, Bareiss E. On roundoff error distributions in floating point and logarithmic arithmetic. Computing 1985;34(4):325–47.