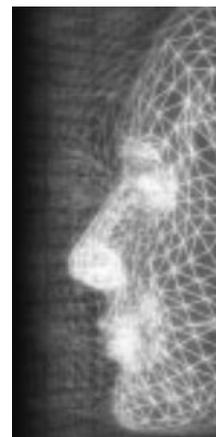


2D shape manipulation via topology-aware rigid grid

By Wenwu Yang and Jieqing Feng*



This paper presents a new method which allows user to manipulate a two-dimensional shape in an intuitive and flexible way. The shape is discretized as a regular grid. User places handles on the grid and manipulates the shape by moving the handles to the desired positions. To meet the constraints of the user's manipulation, the grid is then deformed in an as-rigid-as-possible way. However, this straightforward approach tends to produce unnatural deformations when the grid resolution is not high enough to capture the topological structure of the shape. In the proposed method, the regular grid is trimmed and only the cells that are inside the fatty regions of the shape are preserved, namely "interior grid." When user manipulates the shape, the interior grid and the shape boundary curve are deformed with minimum distortions. To make the deformations of the interior grid and the boundary curve consistent, a junction energy is introduced. In this way, the unnatural deformation effects could be effectively removed and the physically plausible results can be obtained. Meanwhile, the proposed approach provides user an intuitive and simple way to adjust the shape global and local stiffnesses. The deformation is formulated as an energy minimization problem. The energy function is non-quadratic and could be efficiently solved using an iterative solver with the fast summation technique that exploits the interior grid and boundary curve regularities. In addition, the method could be easily extended to manipulate curves and stick figures. Experimental results demonstrate the capability and flexibility of the new method. Copyright © 2009 John Wiley & Sons, Ltd.

Received: 24 March 2009; Accepted: 24 March 2009

KEY WORDS: shape manipulation; topology-aware; tunable stiffness; nonlinear optimization

Introduction

The 2D shape manipulation provides user an intuitive way to move, rotate, stretch, and bend a 2D image or drawing interactively. It has been proven to be an versatile tool in various applications, and is currently prevalent in commercial video processing softwares and vector graphics softwares such as Adobe After Effects[®], Adobe Illustrator[®], as well as cartoon animation softwares such as ToonBoom[®].

Free-form deformation (FFD) and skeleton-based techniques are two prevalent methods for shape manipulation. They manipulate the object by editing the

pre-defined lattice or skeleton. However, manipulating densely distributed control points in the lattice is a laborious work and defining an appropriate skeleton is not trivial especially for the object with an ambiguous joint structure.

Recently, a wide class of methods,^{1–4} which provide users an intuitive interface, have been proposed. These methods allow direct manipulation on an object by editing the specified constraints in a click-and-drag way. Then the shape is deformed locally or globally in a physically plausible way; meanwhile the deformed shape satisfies the constraints as closely as possible. Whilst these methods all aim to minimize the local distortions of the shape interiors during the deformation (i.e., as rigid as possible deformation), one of differences among them is the underlying shape tessellation structure on which the local distortion measurements

*Correspondence to: J. Feng, State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310027, P. R. China.
E-mail: jqfeng@cad.zju.edu.cn; wwyang@cad.zju.edu.cn

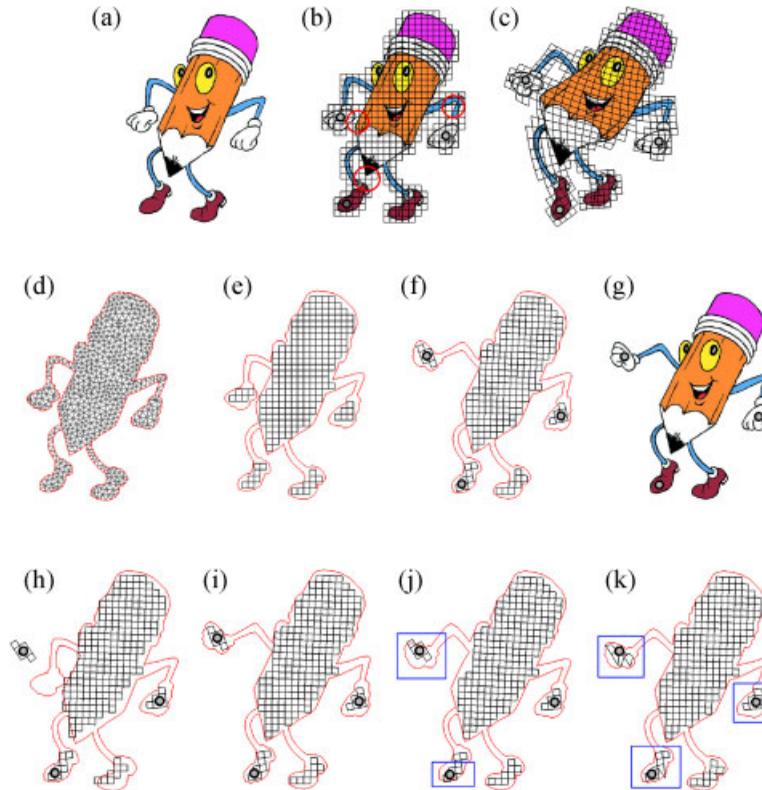


Figure 1. (a–g) Comparison between the straightforward regular grid based deformation method⁴ and our topology-aware method: (a) rest shape; (b) regular grid; (c) unnatural result; (d) triangulation; (e) interior grid; (f) deformed grid; and (g) natural result. Under the same user constraints, the results of Reference [4] and our approach are shown in (c) and (g), respectively. (h–k) Deformations of the interior grid and the boundary curve: (i) $w_s = 1$; (j) $w_s = 0.01$; and (k) $w_s = 100$. (h) Without junction energy. (i–k) With junction energy ($w_r = 6$). The deformed grid cells in the marked squares of (j) and (k) are unmatched with the boundary curve or stretched due to unsuitable w_s .

are defined appropriately. For example, References [1–3] discretize the 2D shape using triangles or quadrangles, however, Reference [4] discretizes the shape using regular grid cells. In the proposed method, a hybrid shape representation is adopted. The hybrid shape representation includes two tessellation structures: one is the regular grid which discretizes the shape fatty regions, namely “interior grid,” as shown in Figure 1(e); the other is the constrained triangulation of the shape interior, as shown in Figure 1(d). Based on this hybrid shape representation, we introduce a new deformation method.

In Reference [4], Yang *et al.* discretized the 2D shape using a regular grid as shown in Figure 1(b) and edited the shape by manipulating the regular grid subject to the user constraints as shown in Figure 1(c). Based on the grid regularity, they designed a stiffness tunable deformation

model and adopted the fast summation technique to speed up the algorithm such that the user could manipulate both soft and stiff objects interactively. But this straightforward method tends to produce unnatural deformation results as shown in Figure 1(c) when the grid resolution is not high enough to capture the shape topological structure (see the parts marked by circles in Figure 1b). One solution is to increase the grid resolution, however, in this way the regular grid scale will increase quadratically and may lead to the interactive editing prohibitive. Another solution is to use the adaptive grid such as Reference [5]. However the adaptive grid structure may make the shape global stiffness adjustment unintuitive since its cells are not in the uniform size. In addition, although the fast summation technique could be extended to the adaptive grid case, it is not a trivial work yet.⁶

On the other hand, the triangular mesh based methods such as References [1,3] do not suffer the resolution problem mentioned above because the triangular mesh, which discretizes the shape interior, captures the shape topological structure fully. In this paper, the proposed method combines the advantages of both the regular grid and the triangular mesh. It constructs a well stiffness-tunable deformation model based on the interior regular grid and eliminates the unnatural results by considering the topological structure described by the triangular mesh, as shown in Figure 1(f) and (g). Meanwhile, by exploiting the interior grid regularity, the proposed method still could be speeded up for several times with the fast summation technique,⁷ especially for the shape with the large global stiffness, making it more suitable for the interactive editing. Furthermore, the proposed method can be easily extended to manipulate the 2D curves and stick figures, which is not trivial for the straightforward regular or adaptive grid based methods such as References [4–6].

Related Work

As a useful tool, the shape manipulation or deformation methods have been well studied for many years. Many algorithms have been introduced and we discuss those that are most relevant to our method.

The prevalent FFD,⁸ skeleton-driven⁹ and other space-warp methods¹⁰ deform an intermediate space where the underlying shape is embedded. These methods are purely geometric approaches and conduct the deformation on intermediate space, thus the generated deformation may be different from the real object behaviors or appearances.

Physical-based approaches such as finite element methods¹¹ can simulate the small-scale object deformation behaviors in a physical accuracy and correctness way. In practical application, the shape manipulations go well beyond the small-scale displacement, however, physical simulation of large-scale deformation is still a challenge problem.

The differential domain methods are also geometry-based approaches.^{12,13} They are not confined to small-scale deformations and can produce physically plausible results. In these settings, surface details are encoded as the differential coordinates and the deformation is formulated as a local rigid transformation optimization problem to preserve the differential coordinates. Similarly, shape matching based methods also preserve

the local shape details, where the local deformations are confined to be rigid transformations.^{3–5} All of them are formulated as a nonlinear optimization problem since neither a 2D nor a 3D rotation transformation can be expressed as a linear function of position coordinates. In Reference [1], this nonlinear optimization was decomposed into two closed-form linear sub-systems. Meanwhile, other linearization methods are also proposed to approximate the rotation transformation.^{12,14} However, these linear approximations may lead to sub-optimal results which correspond to deformation artifacts. The proposed method in this paper falls into the class of nonlinear methods, which can avoid the artifact problem and achieve the high-quality deformations through an iterative optimization.^{3,5}

Algorithm

The method takes as input a 2D shape which is represented as a planar polygon. Then the 2D shape is discretized using a regular grid¹⁵ as shown in Figure 1(b); meanwhile it is triangulated via a Delaunay triangulation subject to boundary and area constraints, as shown in Figure 1(d). The regular grid is then trimmed by the shape boundary, and only the cells in the shape fatty regions are preserved, called “interior grid,” as shown in Figure 1(e). Note that the interior grid cells may be disjointed since some shape skinny regions may contain no grid cells. Meanwhile, the shape boundary curve is refined by inserting points evenly on those edges whose lengths are larger than the grid cell size, such that the average edge length of the new shape boundary curve approaches to the grid cell size.

User manipulates the shape by placing handles on the vertices of the interior grid or(and) the boundary curve and moving them to the desired positions. During the manipulations, the interior grid and the boundary curve are deformed with the least distortions subject to the user’s constraints, i.e., in an as-rigid-as-possible way. Meanwhile, the deformations of the interior grid and the boundary curve are coordinated via a junction energy defined on a junction triangular mesh region. An example of the deformations is shown in Figure 1(f). Finally, the deformations of the interior grid and the boundary curve are transferred to the shape triangulation, and then to the embedded 2D shape, as shown in Figure 1(g).

Local Rigid Regions

The neighboring cells of the interior grid are grouped as local regions. During deformation, these local regions tend to be as rigid as possible, i.e., their distortions are minimized, thus they are called “interior rigid regions.” Similarly, the neighboring edges of the refined boundary curve are grouped as the “boundary rigid regions.” For descriptive convenience, the vertex in the interior grid or the boundary curve is called “node” throughout the rest of the paper whilst the vertex in the triangular mesh remains called “vertex.”

Let \mathcal{L}^{in} be the set of the interior grid nodes. The one-ring neighborhood of an interior grid node consists of those nodes that share with it at least one interior grid cell. Then for each node $i \in \mathcal{L}^{\text{in}}$, an interior rigid region $\mathcal{R}_i^{\text{in}}$ with half-width w is defined as the set of cells that composes of nodes reachable by traversing no more than w -ring neighborhood of the node i . Thus for a fully connected interior grid, an interior rigid region $\mathcal{R}_i^{\text{in}}$ is a square in general and consists of $(2w)^2$ cells (or $(2w + 1)^2$ nodes). Obviously, With the increase of the half-width w , the neighboring rigid regions will be more tightly tied, such that they are more difficult to be stretched or bent. Thus the half-width w can be regarded as a reasonable global stiffness control parameter, i.e., large value for stiff object whilst small value for soft object.

The boundary rigid region is defined on the boundary curve similarly. Let \mathcal{L}^b be the set of the boundary curve nodes. For each node $i \in \mathcal{L}^b$, a boundary rigid region \mathcal{R}_i^b with half-width \tilde{w} is defined as the consecutive $2\tilde{w}$ edges, which consists of the following $2\tilde{w} + 1$ nodes: $i - \tilde{w}, \dots, i, \dots, i + \tilde{w}$. Unlike the half-width w of an interior region, \tilde{w} is determined according to the w . To achieve the consistent deformations of the interior and boundary rigid regions, the length of each boundary rigid region should approach to the width of the interior rigid region. Thus the half-width \tilde{w} is determined by satisfying the equation: $2\tilde{w}l_b \approx 2wl_{\text{in}}$, where l_b is the average edge length of the boundary curve and l_{in} the interior grid cell size. As l_b approaches to l_{in} in our setting, \tilde{w} approaches to w in general.

Deformation Energy

Rigidity. Ideally, the deformations of interior and boundary rigid regions should be rigid transformations. To meet the user’s constraints, the rigid regions can only be deformed in an as-rigid-as-possible way. To achieve this goal, it is important to define distortion metrics for

the two types of rigid regions. Let \mathcal{R}_i be a rigid region at an interior grid node or a boundary curve node i . The initial and deformed positions of a node j are denoted as \mathbf{p}_j and \mathbf{q}_j , respectively. Then we define the distortion metric for the rigid region as

$$E(\mathcal{R}_i) = \sum_{j \in \mathcal{R}_i} \|\mathbf{q}_j - \mathbf{q}_{\text{ic}} - \mathbf{A}_i(\mathbf{p}_j - \mathbf{p}_{\text{ic}})\|^2 \quad (1)$$

where \mathbf{A}_i is the optimal rotation which transforms the rigid region from the initial position to the deformed position, and $\mathbf{p}_{\text{ic}}(\mathbf{q}_{\text{ic}})$ are the initial(deformed) rotation centers. The \mathbf{A}_i can be found by minimizing Equation (1) and there is an analytical solution for 2D case.¹⁶

Since a rigid region is assumed to be translated and rotated as a whole (i.e., as rigid as possible), it is straightforward to choose the initial and deformed rotation centers \mathbf{p}_{ic} and \mathbf{q}_{ic} as follows:

$$\begin{cases} \mathbf{p}_{\text{ic}} = \mathbf{p}_i; \mathbf{q}_{\text{ic}} = \mathbf{q}_i & \text{no node } j \in \mathcal{R}_i \text{ is a handle} \\ \mathbf{p}_{\text{ic}} = \mathbf{p}_s; \mathbf{q}_{\text{ic}} = \mathbf{q}_s & \exists s \in \mathcal{R}_i \text{ is a handle} \end{cases} \quad (2)$$

To achieve as-rigid-as-possible deformation, the total rigid regions distortions should be minimized, i.e., minimizing the following energy function,

$$E_r = E_{\text{rin}} + E_{\text{rb}} = \sum_{i \in \mathcal{L}^{\text{in}}} w_i \frac{E(\mathcal{R}_i^{\text{in}})}{\#\mathcal{R}_i^{\text{in}}} + \sum_{i \in \mathcal{L}^b} w_i \frac{E(\mathcal{R}_i^b)}{\#\mathcal{R}_i^b} \quad (3)$$

where $\#\mathcal{R}_i^{\text{in}}$ and $\#\mathcal{R}_i^b$ denote the number of nodes in the interior rigid region $\mathcal{R}_i^{\text{in}}$ and the boundary rigid region \mathcal{R}_i^b , respectively. Since the interior grid cells are in the same size and the boundary curve nodes are distributed nearly uniformly on the boundary curve, the energy normalization in Equation (3) disperses the energy on the interior or boundary rigid regions evenly. The weight w_i is penalty factor for the average distortion on an interior or boundary rigid region, and hence could be regarded as a local stiffness control parameter.

Junction. Minimizing the energy function Equation (3) will deform the interior grid and the boundary curve independently in an as-rigid-as-possible way. Because there is no explicit connection between them, thus it will lead to the deformations of the interior grid and the boundary curve unmatched, as illustrated in Figure 1(h). This problem can be addressed by adding edge connections between the interior grid and the boundary curve as in Reference [2], but the subsequent relaxing process in the numerical solution will not only

be computationally expensive but also difficult to be implemented. Furthermore, the regular neighborhood structure of the interior grid and the boundary curve will be destroyed, and hence the fast summation,⁷ an efficient technique for accelerating the energy function optimization, could not be employed.

We address this problem by introducing an additional junction energy to coordinate the interior grid deformation and the boundary curve deformation. The junction energy is defined on the gaps between the interior grid and the boundary curve. During deformation, the junction energy ties the interior grid and the boundary curve tightly by preserving the local areas of the gaps, such that their deformations can be coordinated, as shown in Figure 1(i).

Here the key issues are on how to delimit the gap region and how to define the junction energy. As shown in Figure 2, the gaps between the interior grid and the boundary curve can be fully covered by two types of triangles in the triangular mesh. The first type is composed of those that are outside of the interior grid cells (the solid triangles) and the second type is composed of those that intersect the boundary cells of the interior grid (the hatching triangles). For convenience, the two types of triangles are called as “junction triangles,” and we define the junction energy on the junction triangles. To preserve the local areas of the junction triangles, the junction energy measures the distortions of their local properties: the mean value barycentric coordinates of the vertices (i.e., the relative positions among vertices) and the edge lengths, of the junction triangles.

Let \mathcal{V} be the set of all interior vertices in the junction triangles, where an interior vertex in the junction triangles is the one whose one-ring neighboring vertices are all in the junction triangles. An example of the \mathcal{V} is shown in Figure 2, where the interior vertices in the junction triangles are marked as circle dots. Furthermore, let \mathcal{E} be the set of all triangle edges $\{\mathbf{v}_i\mathbf{v}_j\}$ in the junction triangles, and then the junction energy is defined as

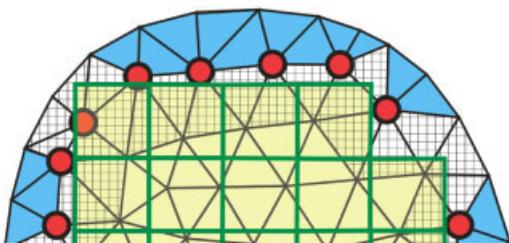


Figure 2. The gaps between the interior grid and the boundary curve.

follows:

$$E_s = \sum_{\mathbf{v}_i \in \mathcal{V}} \left\| \mathbf{v}_i - \sum_{\mathbf{v}_j \in \Omega_i} w_{i,j} \mathbf{v}_j \right\|^2 + \sum_{\mathbf{v}_i \mathbf{v}_j \in \mathcal{E}} \|(\mathbf{v}_i - \mathbf{v}_j) - e(\mathbf{v}_i, \mathbf{v}_j)\|^2 \quad (4)$$

where Ω_i is one-ring neighboring vertices of \mathbf{v}_i , $w_{i,j}$ is the mean value barycentric coordinate of \mathbf{v}_i with respect to the neighbor \mathbf{v}_j , and $e(\mathbf{v}_i, \mathbf{v}_j) = (\tilde{l}_{i,j}/l_{i,j})(\mathbf{v}_i - \mathbf{v}_j)$, where $\tilde{l}_{i,j}, l_{i,j}$ are the original and deformed edge lengths of $\mathbf{v}_i\mathbf{v}_j$, respectively.

Total Energy. To coordinate the interior grid and boundary curve deformations via the junction energy (see Figure 1i), it should disperse the junction energy to the interior grid and the boundary curve such that the local areas of the gaps between them can be preserved during the deformation. To achieve this goal, the junction energy should be reformulated in terms of the interior grid and boundary curve nodes. For this purpose the triangular mesh vertices are classified into three sets: \mathcal{V}^1 —the vertices on the mesh boundary edges, \mathcal{V}^2 —the vertices that are inside the interior grid cells, and \mathcal{V}^3 —the others. Obviously, each mesh boundary vertex in \mathcal{V}^1 can be represented as a linear combination of two boundary curve nodes, and each mesh vertex in \mathcal{V}^2 can be expressed in the bilinear combination of four nodes in the interior grid.

Therefore, for each mesh vertex $\mathbf{v}_i \in \mathcal{V}^{1,2,3}$, its deformation position can be expressed as $\mathbf{v}_i = \mathbf{W}_i \mathbf{Q}_i$, where \mathbf{W}_i is a row vector composed of linear combination coefficients and \mathbf{Q}_i is a column vector composed of 2D position vectors. When $\mathbf{v}_i \in \mathcal{V}^1$, given the deformation positions: \mathbf{q}_1 and \mathbf{q}_2 of the boundary curve nodes that correspond to the end points of the boundary curve edge which the mesh vertex \mathbf{v}_i is on, then $\mathbf{W}_i = (\alpha_1, \alpha_2, 0, 0)$ and $\mathbf{Q}_i = (\mathbf{q}_1, \mathbf{q}_2, \mathbf{0}, \mathbf{0})^T$, where α_1, α_2 are the linear combination coefficients. Note that α_1 or α_2 may be 0 when the mesh vertex \mathbf{v}_i coincides with a boundary curve node. When $\mathbf{v}_i \in \mathcal{V}^2$, we have $\mathbf{v}_i = \alpha_1 \mathbf{q}_1 + \alpha_2 \mathbf{q}_2 + \alpha_3 \mathbf{q}_3 + \alpha_4 \mathbf{q}_4$, where $\{\mathbf{q}_j\}_{j=1}^4$ are the deformed positions of the four nodes of the corresponding interior grid cell in which the mesh vertex \mathbf{v}_i lies, and $\{\alpha_j\}_{j=1}^4$ are the bilinear combination coefficients. Thus $\mathbf{W}_i = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ and $\mathbf{Q}_i = (\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4)^T$. At last, for the $\mathbf{v}_i \in \mathcal{V}^3$, it is just expressed as itself, i.e., $\mathbf{W}_i = (1, 0, 0, 0)$ and $\mathbf{Q}_i = (\mathbf{v}_i, \mathbf{0}, \mathbf{0}, \mathbf{0})^T$. It should be noted that since the vertex $\mathbf{v}_i \in \mathcal{V}^3$ does not lie either in the interior grid cell or on the boundary curve edge, we can draw that the vertex will belong to the junction triangles.

By substituting the expression $\mathbf{v}_i = \mathbf{W}_i \mathbf{Q}_i$ into the junction energy (4), the energy functions E_r and E_s will be described by the common variables, i.e., the deformation positions of all the interior grid and boundary curve nodes: \mathbf{q}_i^{in} and \mathbf{q}_i^b , and the deformation positions of the vertices $\mathbf{v}_i \in \mathcal{V}^3$. Therefore, a uniform energy function can be defined and expressed in terms of the common variables as follows:

$$\min_{\{\mathbf{q}_i^{\text{in}}, \mathbf{q}_i^b\}, \{\mathbf{v}_i \in \mathcal{V}^3\}} w_r(E_{r_{\text{in}}} + E_{r_b}) + w_s E_s \quad (5)$$

In this way the junction energy on the junction triangles is dispersed to the interior grid and the boundary curve. The number of variables in Equation (5) equals to the total numbers of the interior grid nodes and the boundary curve nodes as well as the vertices in \mathcal{V}^3 . In general, the number of vertices in \mathcal{V}^3 is small, e.g., 0–10 in all our examples.

In our experiments, we have tested different values for the weights w_r and w_s , and found that $(w_r, w_s) = (6, 1)$ could produce well-coordinated deformations of the interior grid and the boundary curve, as shown in Figure 1(i). Of course, users can tune the two weights, but should be in a reasonable range. For example, when the value w_s is too small the gaps between the interior grid and the boundary curve could not be well preserved (Figure 1j); when the value w_s becomes too large, the gaps may be preserved strongly so that the interior grid or the boundary curve might be distorted (Figure 1k).

Optimization

The optimal problem (5) is nonlinear as the terms $\{\mathbf{A}_i\}$ and $\{e(\mathbf{v}_i, \mathbf{v}_j)\}$ cannot be expressed as linear combinations of their variables. Thus we can optimize the energy function (5) can be optimized by using an iterative scheme. The optimization is subjected to the position constraints, which are the handles specified interactively on the interior grid or boundary curve nodes. The iteration process works as follows with an initial guess of the deformation positions of the interior grid and boundary curve nodes as well as the triangular mesh vertices, i.e., $\{\mathbf{q}_i^{\text{in}}\}$, $\{\mathbf{q}_i^b\}$, and $\{\mathbf{v}_i \in \mathcal{V}^3\}$. In our implementation, the initial guess is taken as the value at the last manipulation and it works very well for the interactive applications.

(1) Starts with the initial guess $\{\mathbf{q}_i^{\text{in}}\}^{(0)}$, $\{\mathbf{q}_i^b\}^{(0)}$, and $\{\mathbf{v}_i \in \mathcal{V}^3\}^{(0)}$.

- (2) At the k th iteration, compute $\{\mathbf{A}_i\}^{(k)}$ for each interior and boundary rigid region by minimizing the energy function $E(\mathcal{R}_i)$ in Equation (1) using $\{\mathbf{q}_i^{\text{in}}\}^{(k-1)}$ or $\{\mathbf{q}_i^b\}^{(k-1)}$ where $\{\mathbf{A}_i\}^{(k)}$ has the analytical solution;¹⁶ compute each $\{e(\mathbf{v}_i, \mathbf{v}_j)\}^{(k)}$ and $\{w_{ij}\}^{(k)}$ in Equation (4).
- (3) Substitute above $\{\mathbf{A}_i\}^{(k)}$, $\{e(\mathbf{v}_i, \mathbf{v}_j)\}^{(k)}$, and $\{w_{ij}\}^{(k)}$ into the Equation (5), and then minimize the energy function (5) by solving a linear system to determine $\{\mathbf{q}_i^{\text{in}}\}^{(k)}$, $\{\mathbf{q}_i^b\}^{(k)}$, and $\{\mathbf{v}_i \in \mathcal{V}^3\}^{(k)}$.
- (4) If a local minimum of energy function (5) is achieved, stop; otherwise go to step 2.

At each iteration, after substituting the nonlinear terms, $\{\mathbf{A}_i\}^{(k)}$, $\{e(\mathbf{v}_i, \mathbf{v}_j)\}^{(k)}$ and $\{w_{ij}\}^{(k)}$, the nonlinear optimal problem (5) will become linear and can be solved by using a standard linear least squares minimization. By calculating the derivatives of energy function (5) with respect to all variables and setting them to be zero, a linear system is obtained: $\mathbf{M}\mathbf{x} = \mathbf{b}$. The linear system is sparse and the matrix \mathbf{M} only depends on the initial configurations of the interior grid and the boundary curve as well as the triangular mesh. Thus a direct solver with a pre-factorization of the matrix \mathbf{M} could be employed here, in which the matrix \mathbf{M} needs to be factored only once.¹⁷ In this way, the linear system could be solved efficiently.

In fact, after pre-factorization, at each iteration only one simple back-substitution is executed for solving the unknown variables, i.e., \mathbf{x} , and the major runtime cost is the computation of the right side of the linear system, i.e., the vector \mathbf{b} . In our experimental examples, about 94–99% computational cost of the vector \mathbf{b} is spent on the summations of the interior or boundary rigid regions. The summations have the form $\sum_{j \in \mathcal{R}_i} C_{ij}$, where \mathcal{R}_i is an interior or boundary rigid region and C_{ij} is the linear combination of the data attached on each node j of the region \mathcal{R}_i , e.g., $\sum_{j \in \mathcal{R}_i} w_i \mathbf{A}_i \mathbf{p}_j$ or $\sum_{j \in \mathcal{R}_i} w_j \mathbf{A}_j \mathbf{p}_i$ etc. Obviously, the computational cost for the summations will increase quadratically for interior rigid region (linearly for boundary rigid region) with the increase of the rigid region half-width w . Thus the naive calculations will be inefficient for interactively manipulating a stiff object which has a large half-width value w .

Thanks to the regularities of the interior grid and the boundary curve, the above problem could be addressed by using the fast summation technique.⁷ The key idea of the fast summation here is to fully reuse the redundant summations on the interior or boundary rigid regions by exploiting their regularities, and eventually reduce

the computational cost sum of all $\sum_{j \in \mathcal{R}_i} C_{ij}$ to be nearly linear in the number of the interior grid or boundary curve nodes, i.e., independent of the region half-width w . The details of fast summation algorithm can be referred to Reference [7]. Finally, the nonlinear optimization could be accelerated so as to facilitate the interactive manipulation of both soft and stiff objects.

Experimental Results and Discussion

The method has been implemented in a single thread way on a PC with 2.4GHz Intel Core 2 Duo CPU and 2GB RAM. The numerical optimization for all our experimental examples are convergent after 6–9 iterations in general. Table 1 lists some performance statistics data. The runtime costs of the interior grid and boundary curve generations as well as the constrained triangulation are not included as they are all less than 3 millisecond for all examples in this paper, and thus could be negligible. From Table 1, we can see that with the increase of the region half-width w , the number of non-zero elements in the sparse linear system will increase accordingly, and the performance of the sparse linear system solver drops either.¹⁷ However by employing the fast summation technique the optimization runtime can still meet the requirement of interactively manipulating a stiff model in the moderate size whilst it is prohibitive without employing the fast summation technique.

Figure 1(g) shows the deformation results using our method. Obviously, our method effectively avoids the topology-unaware results, as shown in Figure 1(c), which is generated by the straightforward regular grid based method.⁴ Figure 1(h–i) shows the importance of the junction energy in Equation (4) which coordinates the interior grid and boundary curve deformations. Even

for the interior grid in the disjointed shape fatty regions, as shown in Figure 1(e), the junction energy can still coordinate the deformations between each part of the interior grid and the boundary curve well (see Figure 1f and g).

The proposed method can generate natural and physically plausible results with a few constraints due to its nonlinear inherence. Figure 3(a–c) shows some natural deformations of a non-articulated 2D snake by manipulating only two constraint handles. In Figure 3(d–g), a 2D bottle is edited by manipulating only four handles.

These deformation effects are comparable to those generated by the two-step linear approximation method,¹ as well as the nonlinear method.² However, as the authors described, the two-step linear approximation method¹ may produce physically implausible results, as shown in Figure 4(b). Our nonlinear method effectively avoids the implausible effects and tends to generate the physically plausible ones as illustrate in Figure 4(c). Furthermore, our method sometimes can generate better results than the nonlinear method,² because the energy based on the shape matching, i.e., the Equation (1) is more suitable to preserve the shape interior rigidity (i.e., internal resistance to deformation) than the energy based on the local shape area preservation in Reference [2], as shown in Figure 4(d–f).

The proposed method can also provide user an intuitive and convenient way to tune the shape global and local stiffnesses. The examples in Figures 3(d–g) and 4(c) and show the deformation results of the shapes with different global stiffnesses.

Curve and Stick Figure Editing

Besides 2D shapes, the proposed method can also be applied to other 2D objects, such as planar curves or

Figure	VER	NOD	w	PRE	OPT1	OPT2
3(a–c)	904	793	3	11.2	10.7	17.5
3(d–g)	3227	2256	1;6;9	37.1;175.6;293.7	44.3;96.5;114.2	45.4;256.1;467.1
4(c)	755	570	1;3;6	7.3;11.5;16.1	8.1;10.7;12.8	9.3;18.6;36.5
4(f)	592	512	3	8.5	7.5	12.0

Table 1. Performance statistics

The runtime cost is measured in millisecond. VER: number of triangular mesh vertices; NOD: number of the interior grid and boundary curve nodes; w: region half-width; PRE: runtime of pre-computation; OPT1/OPT2: runtime of iterations with/without fast summation technique.

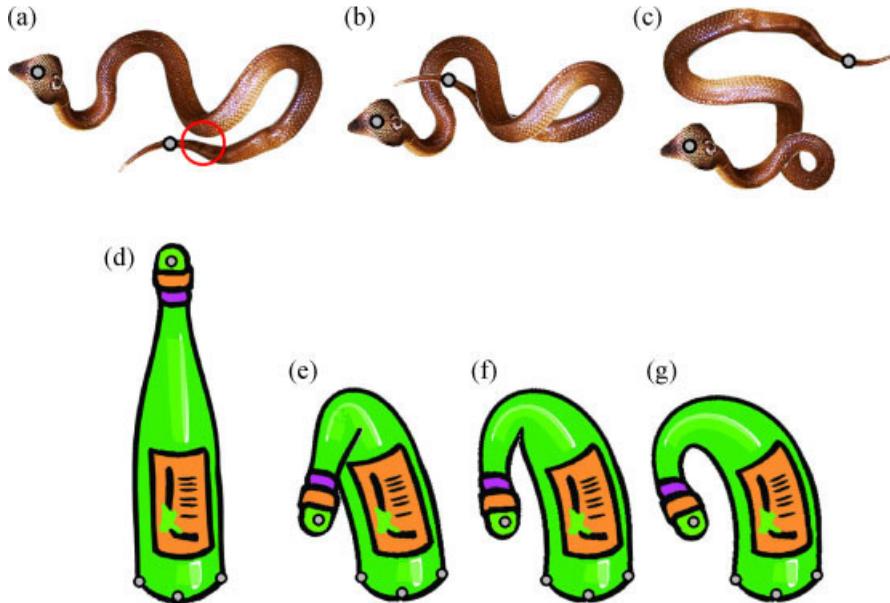


Figure 3. (a–c) Manipulation of a non-articulated 2D snake with two handles: (a) the rest shape. (d–g) Deformations of a 2D bottle with the increasing global stiffness from left to right, where $w = 1, 6, 9$, respectively: (d) the rest shape.

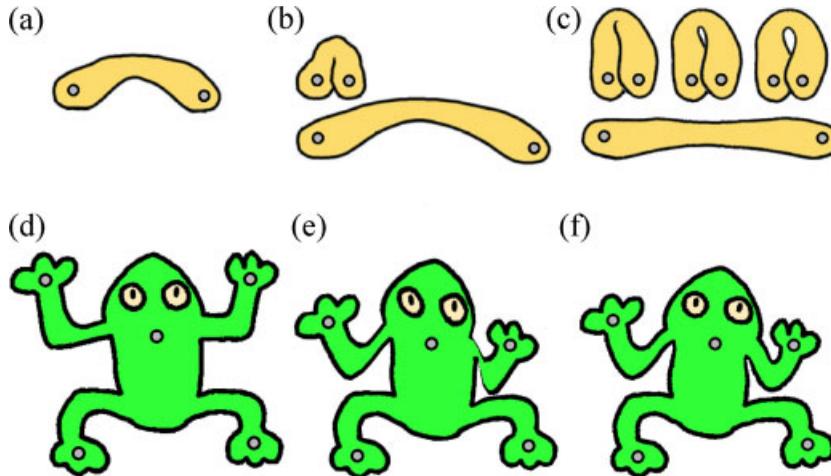


Figure 4. Comparison between (a) the rest shape, (b) the two-step linear approximation method¹, and (c) our method. The upper row of (c) shows the shape deformations with the different global stiffness parameters, i.e., $w = 1, 3, 6$, respectively. Comparison between (d) the rest shape (e) the nonlinear method², and (f) our approach. In (e) the shrinkage occurs at the frog right arm.

stick figures etc. A planar curve could be regarded as a 2D shape without interior part. Therefore, the proposed method applied to the planar curve editing could be simplified by only optimizing the energy function E_r , in Equation (3). Figure 5(a–b) show an example of the spring-like curve deformations with different global

stiffnesses, in which the curve details are well preserved and the “spring” deformation results mimic well the real spring deformations.

A stick figure is defined as a connected planar graph, which is composed of open and close 2D shapes as shown in Figure 5(c). The proposed method could also be

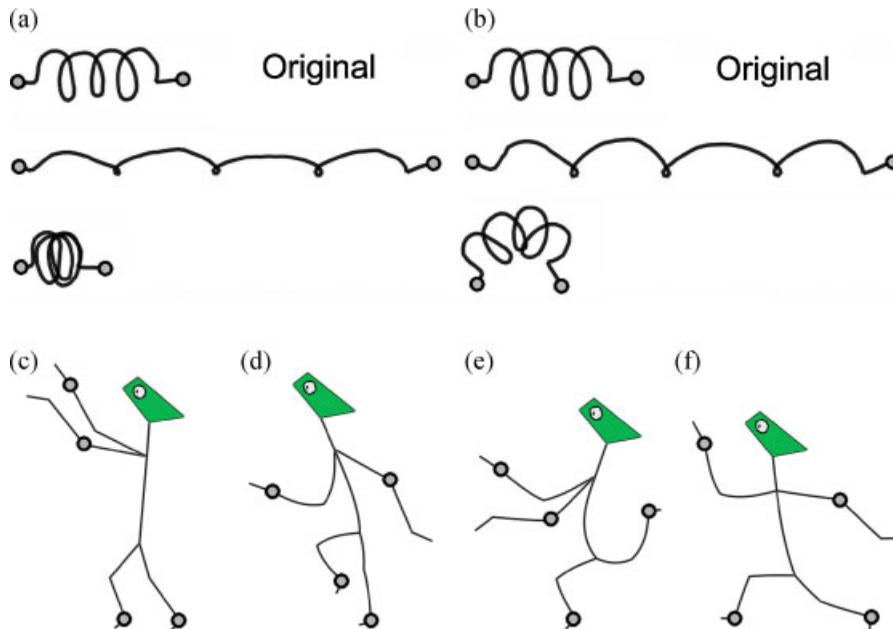


Figure 5. (a–b) Curve editing with different global stiffnesses: (a) soft ($w = 1$) and (b) stiff ($w = 6$). They mimic the behaviors of stretching and squashing a soft and stiff springs in a physically plausible way. (c–f) An example of the stick figure manipulation: (c) the rest figure.

applied to the stick figure deformation straightforwardly. Whilst the boundary rigid regions are defined on the refined edges of both open and close shapes, the interior rigid regions are only defined on the interior grids of the close shapes. The energy functions E_{rb} , E_{rin} , and E_s could be defined accordingly, and a total energy function for the stick figure manipulation is as follows:

$$E_f = \sum_{i=1}^n w_r E_{rb}^i + \sum_{i=1}^m w_r E_{rin}^i + w_s E_s^i \quad (6)$$

where n is the number of both the open and close shapes, and m is the number of the close ones in the stick figure. Figure 5(c–f) show an example of the stick figure manipulation.

Conclusion and Future Work

This paper presents a two-dimensional shape deformation method via direct manipulation. It adopts a hybrid shape representation structure which composes of an interior regular grid and a triangular mesh. The approach constructs a global and local stiffness-tunable

deformation model based on the interior regular grid and eliminates the unnatural results by considering the topological structure implied in the triangular mesh. Meanwhile, by exploiting the interior grid and boundary curve regularities, the method can be speeded up using the fast summation technique so as to manipulate both the soft and stiff objects interactively. Furthermore, the proposed method can be extended to manipulate the 2D curves and stick figures.

As the future work, the proposed method will be extended to 3D case. However, it is not a trivial work. Compared with the 2D triangular mesh, the 3D tetrahedralization of a mesh is more difficult to be implemented and computationally expensive, as well as is not robust. Therefore, the key step in 3D case is to construct a suitable tessellation structure that is able to capture the 3D surface topology.

ACKNOWLEDGEMENTS

The work was jointly supported by the NSF of China (60743002), the 973 Program of China (2009CB320801), the National Key Technology R&D Program (2007BAH11B02), and the NSF of Zhejiang Province (R106449).

References

1. Igarashi T, Moscovich T, Hughes JF. As-rigid-as-possible shape manipulation. *ACM Transactions on Graphics* 2005; **24**(3): 1134–1141.
2. Weng Y, Xu W, Wu Y, Zhou K, Guo B. 2d shape deformation using nonlinear least squares optimization. *The Visual Computer* 2006; **22**(9): 653–660.
3. Sorkine O, Alexa M. As-rigid-as-possible surface modeling. In *SGP'07: Proceedings of the 5th Eurographics Symposium on Geometry Processing*, July 2007; 109–116.
4. Yang W, Feng J, Jin X. Shape deformation with tunable stiffness. *The Visual Computer* 2008; **24**(7–9): 495–503.
5. Botsch M, Pauly M, Wicke M, Gross M. Adaptive space deformations based on rigid cells. *Computer Graphics Forum* 2007; **26**(3): 339–347.
6. Steinemann D, Otaduy MA, Gross M. Fast adaptive shape matching deformations. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2008; 87–94.
7. Rivers AR, James DL. Fastlsm: fast lattice shape matching for robust real-time deformation. *ACM Transactions on Graphics* 2007; **26**(3): 82.
8. Sederberg TW, Parry SR. Free-form deformation of solid geometric models. *Computer Graphics (SIGGRAPH 1986)* 1986; **20**(4): 151–160.
9. Lewis JP, Cordner M, Fong N. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH'00*, July 2000; 165–172.
10. Botsch M, Kobbelt L. Real-time shape editing using radial basis functions. *Computer Graphics Forum* 2005; **24**(3): 611–621.
11. Schiwietz T, Georgii J, Westermann R. Freeform image. In *Proceedings of Pacific Graphics 2007*, October 2007; 27–36.
12. Sorkine O, Cohen-Or D, Lipman Y, Alexa M, Rössl C, Seidel H-P. Laplacian surface editing. In *SGP'04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, July 2004; 175–184.
13. Yu Y, Zhou K, Xu D, Shi X, Bao H, Guo B, Shum H-Y. Mesh editing with poisson-based gradient field manipulation. *ACM Transactions on Graphics* 2004; **23**(3): 644–651.
14. Lipman Y, Sorkine O, Levin D, Cohen-Or D. Linear rotation-invariant coordinates for meshes. *ACM Transactions on Graphics* 2005; **24**(3): 479–487.
15. James DL, Barbič J, Twigg CD. Squashing cubes: automating deformable model construction for graphics. In *SIGGRAPH'04: ACM SIGGRAPH 2004 Sketches*, August 2004; 38.
16. Schaefer S, McPhail T, Warren J. Image deformation using moving least squares. *ACM Transactions on Graphics* 2006; **25**(3): 533–540.
17. Davis TA. UMFPACK, an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software* 2004; **30**(2): 196–199.

Authors' biographies:



Wenwu Yang is a doctoral student in the State Key Lab of CAD&CG, Zhejiang University, Peoples Republic of China. He received his MSc in computer graphics from Zhejiang University in 2005. His research interests include computer graphics, geometric modeling, and computer animation.



Jieqing Feng is a professor in the State Key Lab of CAD&CG, Zhejiang University, Peoples Republic of China. He received his BSc in applied mathematics from the National University of Defense Technology in 1992 and his PhD in computer graphics from Zhejiang University in 1997. His research interests include geometric modeling, rendering, and computer animation.