Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Three-dimensional deformation in curl vector field*

Dan ZENG[1], Da-yue ZHENG[‡1,2]
([1]*Key Laboratory of Specialty Fiber Optics and Optical Access Networks, Shanghai University, Shanghai 200072, China*)
([2]*NetEase (Hangzhou) Network Co., Ltd., Hangzhou 310052, China*)
E-mail: dzeng@shu.edu.cn; iwalu1999@hotmail.com
Received Jan. 9, 2012; Revision accepted May 11, 2012; Crosschecked July 6, 2012

Abstract: Deformation is an important research topic in graphics. There are two key issues in mesh deformation: (1) self-intersection and (2) volume preserving. In this paper, we present a new method to construct a vector field for volume-preserving mesh deformation of free-form objects. Volume-preserving is an inherent feature of a curl vector field. Since the field lines of the curl vector field will never intersect with each other, a mesh deformed under a curl vector field can avoid self-intersection between field lines. Designing the vector field based on curl is useful in preserving graphic features and preventing self-intersection. Our proposed algorithm introduces distance field into vector field construction; as a result, the shape of the curl vector field is closely related to the object shape. We define the construction of the curl vector field for translation and rotation and provide some special effects such as twisting and bending. Taking into account the information of the object, this approach can provide easy and intuitive construction for free-form objects. Experimental results show that the approach works effectively in real-time animation.

Key words: 3D mesh deformation, Curl vector field, Volume preserving, Self-intersection
doi:10.1631/jzus.C1200004 **Document code:** A **CLC number:** TP391

1 Introduction

Shape deformation is one of the most important research fields in computer graphics. Development of this popular aspect of geometry processing diffusely affects computational geometry, simulation, computer-aided design (CAD), and animation. Based on the common requests of users, there are mainly two core problems involved in this key research field: non-self-intersecting and volume preserving. In addition, smoothness-preserving and easy user-interaction are generally required. To satisfy these constraints, several advanced algorithms are developed, including the well-known free-form deformation (FFD) (Sederberg and Parry, 1986), the extended manifold method, Laplacian deformation (Sorkine *et al*., 2004), and vector field deformation.

Vector field deformation is applied not only to shape deformation but also to texture synthesis, non-photorealistic rendering, and fluid simulation (Zhang *et al*., 2006). Davis (1967) introduced an important property of the divergence-free vector field—deformation under the divergence-free vector field is volume-preserving. Theisel *et al*. (2005) mentioned that non-self-intersection occurs in the divergence-free vector field since the field path lines have no intersection with each other in the domain. von Funck *et al*. (2006) demonstrated that divergence-free vector field deformation is C^1 continuous, which means that it is smoothness-preserving. Such deformation also preserves detailed features. It is obvious that the properties of the divergence-free vector field are quite suitable for use in deformation.

von Funck *et al*. (2006) also provided an innovative algorithm to construct a divergence-free vector field for volume-preserving deformation. The basic

‡ Corresponding author
* Project (Nos. 40905013 and 60832003) supported by the National Natural Science Foundation of China, the Shanghai Natural Science Foundation (No. 11ZR1413400), the Key Scientific Research Project of the Shanghai Education Committee (No. 12YZ007), and the Open Project Program of the State Key Lab of CAD&CG, Zhejiang University (No. A1101), China

idea is that a divergence-free vector field can be constructed from the cross product of the gradients of two scalar fields:

$$\boldsymbol{v}(x,y,z)=\nabla p(x,y,z)\times\nabla q(x,y,z). \quad (1)$$

This provides a spherical vector field primitive for dragging and a cylindrical vector field primitive for bending.

The design of vector field primitives (von Funck *et al*., 2006) is, however, based on pre-defined shapes, and is not related to object geometry. While this kind of design may work well on spherical or cylindrical objects, it does not work on free-form objects. The contribution of this paper is introducing a new method of divergence-free vector field primitive construction for general shapes. The spherical vector field primitive (von Funck *et al*., 2006) is addressed as a special case in our method.

The new method is based on the distance field rather than pre-defined shapes, so the object's geometric information will be used in vector field design. Fig. 1 shows a vector field deformation example using our method.

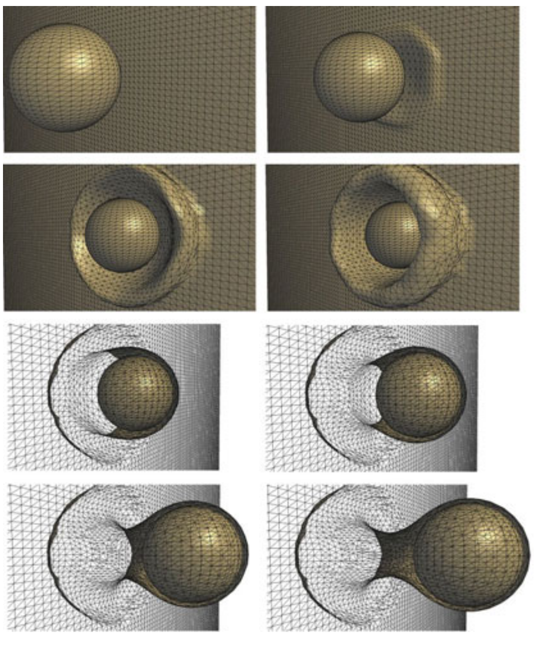

Fig. 1 A curl vector field deformation example
When the sphere keeps moving, the curl vector field around the sphere occurs, which results in volume-preserving deformation on the surface in real time, without GPU speed-up technology

2 Related work

Shape deformation, especially 3D deformation, is a hot research field in computer graphics. Early surface deformation approaches can be classified according to their control handles. Sederberg and Parry (1986) introduced the famous lattice free-form deformation (FFD) technique. It was extended by Coquillart (1990) using nonparallel-epipedical lattices. While the lattice handle is good at global shape control, it has problems with local detail adjustment. Thus, new algorithms based on wire or vertex handles are developed to deal with this problem. Barr (1984) and Singh and Fiume (1998) exploited new deformable proxy constructed by wire handles with domain curves in shape deformation. For accurate manipulation, Hsu *et al*. (1992) and Hirota *et al*. (2000) defined multi-level boundary representations for modeling primitives, and developed an avertex-based controlling algorithm from the lattice-based FFD algorithm.

Generally speaking, classical FFD methods and their variants usually require complex user-interaction (Botsch and Kobbelt, 2004). More advanced deformation methods are developed to simplify the controlling process and to satisfy given constraints by deigning different basis functions. Botsch and Kobbelt (2005) used triharmonic radial basis functions for all kinds of scattered data interpolation problems, while Angelidis *et al*. (2004) defined a new shape deformation tool named swirling-sweeper based on their framework. It is based on blending several rotations whose magnitude decreases away from the control center.

The Laplacian approach has been well developed with respect to shape deformation since Laplacian coordinates represent surface details using the local mean (Alexa, 2003; Lipman *et al*., 2004). Sorkine *et al*. (2004) discussed surface editing over an intrinsic surface representation of a surface. Zhou *et al*. (2005) extended the idea to the volumetric domain to solve the problem of large deformations. Vector field analysis has been widely used in 2D/3D shape deformation, fluid simulation (Stam, 2003), visualization (van Wijk, 2003), and texture synthesis (Praun *et al*., 2000). Zhang *et al*. (2006) presented a vector field design system to create a wide variety of vector fields that allow the user to control over vector field topology. von Funck *et al*. (2006) defined a new shape deformation method based on time-dependent divergence-free vector fields. The deformation holds such prosperities as self-intersection-free, volume-preserving, smoothness-preserving, and feature-preserving. As this algorithm constructs only a spherical or cylindrical shape vector field, we extend the idea to free-form shape objects, making this method more flexible for diverse applications. Angelidis and Singh (2007) developed a volume-preserving skinning algorithm based on a powerful embedding into the volumetric space. Rohmer *et al*. (2009) presented a novel method for exact volume-preserving skinning with shape control, which not only offers an exact control of the object volume, but also enables the user to specify the shape of volume-preserving deformations through intuitive 1D profile curves.

3 Curl vector field construction

The goal of the approach is to effectively construct a special vector field from the curl of a base vector field to provide non-self-intersecting and volume-preserving deformation. In general, a vector field deformation involves two objects: deforming objects and deformed objects. The deforming object providing shape information is the base of divergence-free vector field construction.

The deformed object is the object being deformed under the vector field of the deforming object. For some special cases, vector fields can be generated based on pre-defined conditions without any deforming object. However, the deformed object is always required.

Mathematically, a divergence-free vector field has some special features. One of them is that the objects deformed under the divergence-free vector field are volume preserving, which is a very important feature in mesh deformation and is the basic criterion of our algorithm. From previous research, we know that the divergence-free vector field can be constructed from the gradients of two scalar fields $p(x, y, z)$ and $q(x, y, z)$ in the 3D space, or constructed as a co-gradient field of a scalar field $p(x, y)$ in the 2D space (von Funck *et al*., 2006). The construction is based on a well-known feature—the cross product of the gradients of two scalar fields is divergence-free (Davis, 1967).

Fig. 2 illustrates the vector flow in the curl vector field. Fig. 3 gives the comparison of different divergence-free vector field construction algorithms.

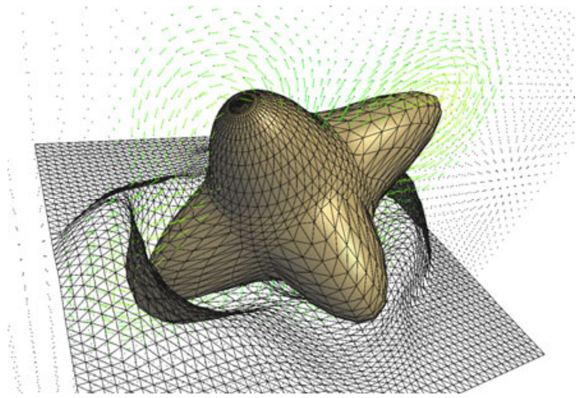

Fig. 2 Illustration of vector flow in the curl vector field
The colored vectors on the section plane show the directions and magnitude of curl vector flow. The curl vector field has the same shape feature as the shape feature of the deforming object

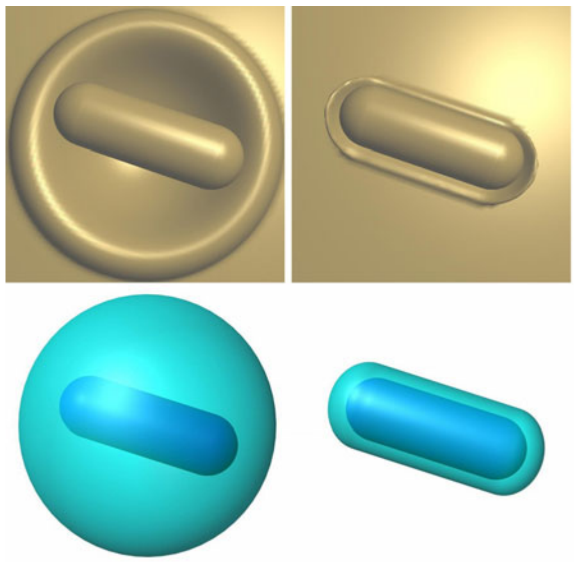

Fig. 3 Comparison of different divergence-free vector field construction algorithms
The second row illustrates the shape of the vector fields. The left column is constructed using an algorithm proposed by von Funck *et al*. (2006), which relates only to a pre-defined field radius; the right column is constructed using the new algorithm, which relates to the distance field of the deforming object

Here, we present a new way of construction, which is more flexible and can be easily understood for 3D divergence-free vector field construction from the curl field. It is also well-known that the divergence of a curl field is equal to zero:

$$\mathrm{div}(\mathbf{curl}(\boldsymbol{v})) = \nabla \cdot (\nabla \times \boldsymbol{v}) = 0. \tag{2}$$

The above feature ensures that our vector field construction based on curl is volume preserving.

3.1 Translation

As shown in Fig. 4, first we construct a base vector field for translation. A local coordinate framework (***i***, ***j***, ***k***) is built around the deforming object for vector field construction. ***i***, ***j***, ***k*** are three orthogonal unit vectors. ***i*** denotes the moving direction of the deforming object when ***j*** and ***k*** are freely chosen.

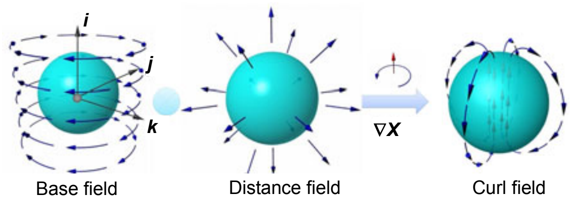

Fig. 4 Illustration of curl vector field generation for translational motion

Vertex $\boldsymbol{X}$ will be denoted as $\boldsymbol{X}'$ in the local coordinate system. For each deforming object, we define a base vector field $\boldsymbol{B}_\text{T}$ rotating around its ***i*** axis in the local coordinate. Let ***P*** be (0, *y*, *z*). We have

$$\boldsymbol{B}_\text{T}(\boldsymbol{X}) = \boldsymbol{P} \times \boldsymbol{i}. \tag{3}$$

We introduce the distance field $d(x, y, z)$ as one of the components in our vector field construction to represent the shape. The value of such a field is the distance to the object surface that separates the inner and outer regions. The deformed object should move only in the outer region during deformation. The distance field is negative in the inner region when it is positive in the outer region. Obviously, the value of the distance field will be zero on the object surface. We want our constructed vector field to affect only the adjacent space of the deforming object and not the whole space. Hence, we denote a field threshold R: when the absolute distance $d(x, y, z)$ is larger than R, then no vector field exists. The blending in the base field will ensure the smoothness of the curl field.

Therefore, the formulation of the divergence-free vector field for translation $\boldsymbol{V}_\text{T}$ is

$$\boldsymbol{V}_\text{T}(x) = \mathbf{curl}(f(x)), \tag{4}$$

where

$$f(x) = \begin{cases} (R - |\,\text{dist}\,|)^2 \boldsymbol{B}_\text{T}(X), & |\,\text{dist}\,| \le R, \\ 0, & |\,\text{dist}\,| > R. \end{cases}$$

Under the definition above, the divergence-free vector field for translation $\boldsymbol{V}_\text{T}$ flows out from the front of the deforming object, and then flows along the surface of the object and goes back into the object at the back (Fig. 5). Compared with von Funck *et al.* (2006), such a definition will have a similar performance for spherical objects. However, this method will work better for free-form objects, because our new construction contains shape information, while the vector field in von Funck *et al.* (2006) is defined only over a spherical region. von Funck *et al.* (2006) used an array of spherical vector fields to simulate the field for general objects. Our new approach, on the other hand, provides a direct definition for free-form objects such that an array of simple vector fields is not required.

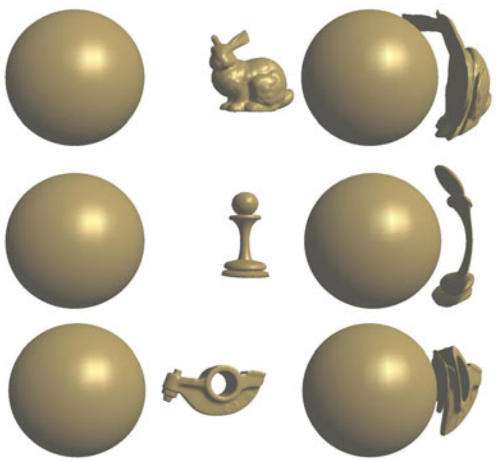

Fig. 5 Three collision examples
From top to bottom: bunny, chess, and rocker-arm

3.2 Rotation

Assume the deforming object rotates around a pivot ***o*** about an **axis**. We can define the curl vector field for rotational motion as follows:

$$\begin{cases} \boldsymbol{V}_\text{R}(\boldsymbol{X}) = \mathbf{curl}(f(x)), \\ \phi(\boldsymbol{X}) = (R - |\,\text{dist}\,|)^2 \boldsymbol{B}_\text{R}(\boldsymbol{X}), & |\text{dist}| \le R, \\ f(\boldsymbol{X}) = \mathbf{0}, & |\text{dist}| > R, \\ \boldsymbol{B}_\text{R}(\boldsymbol{X}) = \boldsymbol{X}' \times \mathbf{axis} \times \boldsymbol{X}, \end{cases} \tag{5}$$

where **axis** is a unit vector in the rotational axis direction.

Fig. 6 shows how the curl vector field is generated. Fig. 7 shows the generated rotational field and its deformation effect for a rod-like object with the rotational axis passing through the center.

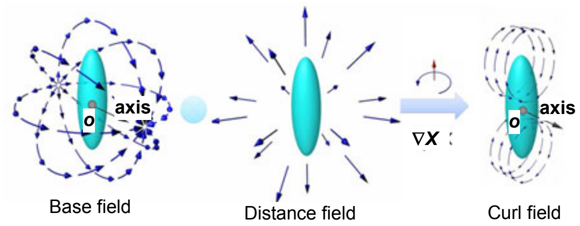

Fig. 6 Illustration of curl vector field generation for rotational motion
o is the rotational pivot and **axis** is the rotational axis

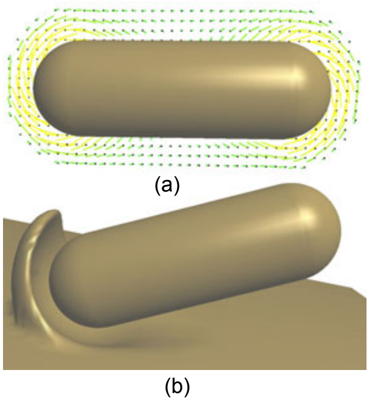

Fig. 7 The vectors of the field (a) and a collision effect between a rotating rod and a surface (b)

3.3 Special application: twisting and bending

Besides translation and rotation, we can develop special applications with curl vector field construction. These kinds of construction result in useful effects for object deformation: twisting and bending.

Fig. 8 and Eq. (6) show the construction of a simple rotational field that can be used for twisting and bending.

$$\boldsymbol{v}(x)=\mathbf{curl}(\boldsymbol{X}'\times\mathbf{axis}\times\boldsymbol{X}). \tag{6}$$

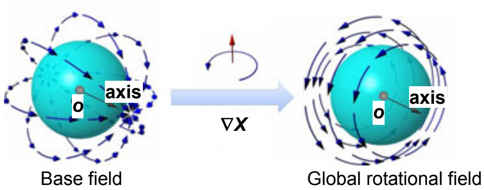

Fig. 8 Simple rotational field generation

We construct the curl vector field of twisting effect by introducing vector projection length L_{axis} on the rotational axis as the constraint parameter in Eq. (7):

$$\boldsymbol{v}(x)=\mathbf{curl}(L_{\text{axis}}\boldsymbol{X}'\times\mathbf{axis}\times\boldsymbol{X}). \tag{7}$$

Fig. 9 shows a smooth twisting effect on a rectangular rod. Fig. 10 shows how twisting is applied on the Venus model.

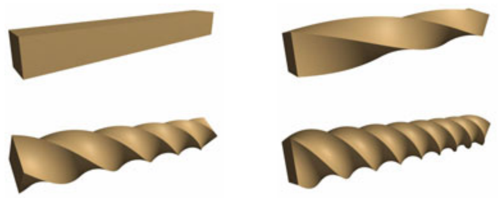

Fig. 9 Rectangular rod in the curl vector field of twisting effect

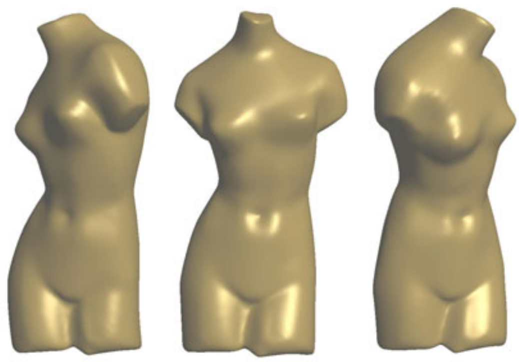

Fig. 10 Human body action: waist twisting of Venus

The construction based on bending effect is similar to the construction using twisting effect. We use the square of the vector length parameter, instead of vector projection length on the rotational axis, as the constraint parameter:

$$\boldsymbol{v}(x)=\mathbf{curl}(|\boldsymbol{X}|^2\cdot\boldsymbol{X}'\times\mathbf{axis}\times\boldsymbol{X}). \tag{8}$$

Fig. 11 shows how the bending effect works on a rectangular rod. Fig. 12 is a Tai Chi image generated based on bending effect.

Fig. 11 Cubes bending effect generated based on the curl vector field

Fig. 12 Tai Chi generated by the curl vector field of bending effect

4 Distance field computation

Distance field is an important component in our divergence-free vector field construction. However, distance field computation is time-consuming. To deal with these problems, we incorporate some ideas from Ohtake *et al*. (2003) to create an implicit function for the domain. This allows for an accurate reconstruction of sharp features and fast local domain access. To perform real-time deformation we also use the classic marching cube algorithm (Lorensen and Cline, 1987) with tri-linear interpolation to speed up our distance field computation.

Keeping the appropriate vertex density of surface is a basis for feature preserving in deformation. Immoderate deformation will cause mesh fragmentation and volume change. We apply remeshing in our algorithm based on ideas from Gain and Dodgson (1999) and von Funck *et al*. (2006). We set several thresholds on triangle edges. When the length of an edge is over the threshold, a 1-to-4 split will occur at all the ring triangles of the related vertex, or 2-to-4 splits will occur at two nearby triangles of long edges, depending on their topological conditions. We also notice that the original mesh and deformation path can be stored for further processing. When ring triangles of vertices or nearby triangles of edges require splits, we can apply splits to the stored original mesh and then integrate the new vertices through the deformation path. This operation has smoother remeshing effect than direct remeshing (von Funck *et al*., 2006).

5 Experimental results

All of the test cases were run on a PC with an Intel Xeon 2.66 GHz CPU and 2.00 GB RAM. Our algorithm has smooth performance in real time without GPU speed-up.

The volume is preserved under the deformation of the curl vector field (Table 1). Because we use triangular mesh objects in our experiments, tiny errors exist in volume computation. There is about a 0.3% difference between the object volumes before and after deformation.

Table 1 Volume-preserving results under the deformation of the curl vector field

Deformation object	Tris	$volume_o$	$volume_d$	F.rate
Sphere (C)	11 200	2.132	2.127	78
Bonny (C)	23 380	1.997	1.988	36
Chess (C)	23 384	0.576	0.572	36
Rocker-arm (C)	43 468	0.143	0.142	30
Cube (R)	33 600	8.000	8.033	39
Venus (R)	36 396	74.811	74.324	37

Tris records the number of object triangles; $volume_o$ and $volume_d$ record the volume data before and after deformation, respectively; F.rate records the average frame rate in deformation. The objects with ‘C’ in brackets are in colliding deformation while the others are in rotating deformation

Since the time complexities of our method and the algorithm based on vector field primitives (von Funck *et al*., 2006) are both $O(N)$, both of them can perform in real time without GPU speed-up. Rather than being related to object geometry, the design of vector field primitives is based on pre-defined shapes, such as a cylinder. This kind of design may work well on spherical or cylindrical objects as shown in Fig. 1, but not on free-form objects such as the feet and hands. In contrast, our method works well on general shapes, as demonstrated in Figs. 13 and 14.

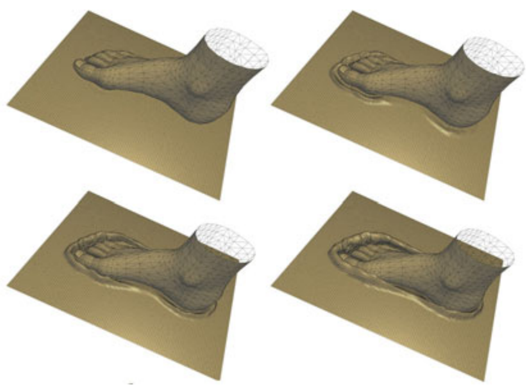

Fig. 13 Deformation of complicated objects: foot print on earth

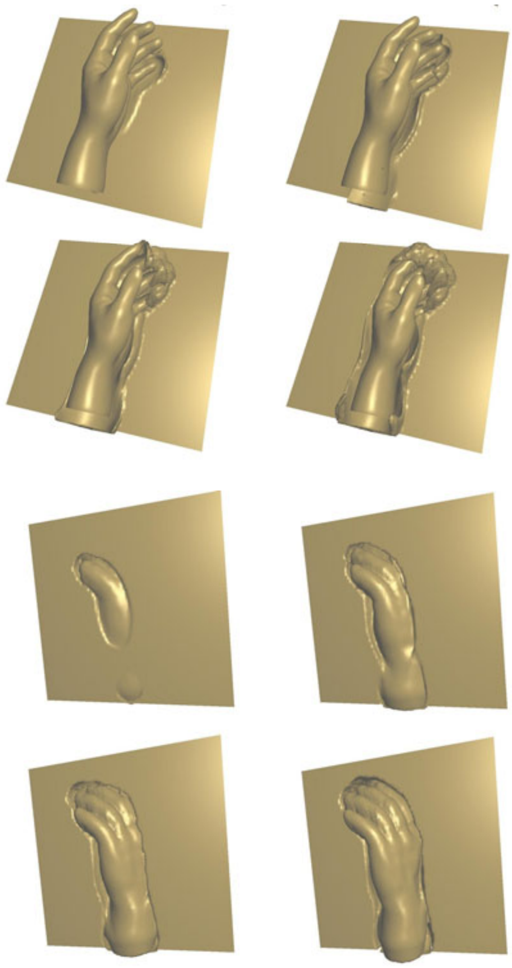

Fig. 14 Deformation of complicated objects: hand print on earth

6 Discussion

Compared with other classical deformation methods, the curl vector field deformation method has many advantages. Volume-preserving is an inherent feature of the curl vector field, because the field lines of the curl of a basic vector field will never intersect with each other, and thus a mesh deformed under the curl vector field can avoid self-intersection. Compared with the method presented by von Funck *et al*. (2006), the new construction is much easier to understand. This is because the construction in von Funck *et al*. (2006) requires the cross product of two vector components, while the new construction using our proposed method is generated directly from a single curl field.

7 Conclusions

In this paper, we have presented a new method to construct the vector field for 3D mesh deformation. The object deformation under the curl vector field is volume-preserving, and there is no self-intersection problem in such deformation due to the properties of the vector field. The algorithm introduces the distance field into vector field construction, so the shape of the curl vector field is closely related to object shape. The construction has a simple mathematical expression and is easy to understand. We have defined the construction of the curl vector field for translation and rotation. Some special effects are provided such as twisting and bending. Experimental results demonstrate that our algorithm is fast enough in performing a real-time deformation without GPU speed-up technology. Our future work will focus on disclosing the possibility of implementation on GPU especially under multiple vector field conditions.

References

Alexa, M., 2003. Differential coordinates for local mesh morphing and deformation. *Vis. Comput.*, **19**(2):105-114.

Angelidis, A., Singh, K., 2007. Kinodynamic Skinning Using Volume-Preserving Deformations. Proc. ACM SIGGRAPH/Eurographics Symp. on Computer Animation, p.129-140.

Angelidis, A., Cani, M.P., Wyvill, G., King, S., 2004. Swirling Sweepers: Constant-Volume Modeling. Proc. 12th Pacific Conf. on Computer Graphics and Applications, p.10-15. [doi:10.1109/PCCGA.2004.1348329]

Barr, A.H., 1984. Global and Local Deformations of Solid Primitives. Proc. 11th Annual Conf. on Computer Graphics and Interactive Techniques, p.21-30. [doi:10.1145/800031.808573]

Botsch, M., Kobbelt, L., 2004. An Intuitive Framework for Real-Time Freeform Modeling. Proc. 31st Int. Conf. on Computer Graphics and Interactive Techniques, p.630-634. [doi:10.1145/1186562.1015772]

Botsch, M., Kobbelt, L., 2005. Real-time shape editing using radial basis functions. *Comput. Graph. Forum*, **24**(3):611-621. [doi:10.1111/j.1467-8659.2005.00886.x]

Coquillart, S., 1990. Extended Free-Form Deformation: a Sculpturing Tool for 3D Geometric Modeling. Proc. 17th Annual Conf. on Computer Graphics and Interactive Techniques, p.187-196. [doi:10.1145/97879.97900]

Davis, H., 1967. Introduction to Vector Analysis. Allyn and Bacon Inc., Boston, USA.

Gain, J.E., Dodgson, N.A., 1999. Adaptive Refinement and Decimation under Free-Form Deformation. Proc. 17th Eurographics, p.13-15.

Hirota, G., Maheshwari, R., Lin, M.C., 2000. Fast volume-preserving free-form deformation using multi-level optimization. *Comput.-Aid. Des.*, **32**(8-9):499-512. [doi:10.1016/S0010-4485(00)00038-5]

Hsu, W.M., Hughes, J.F., Kaufman, H., 1992. Direct Manipulation of Free-Form Deformations. Proc. 19th Annual Conf. on Computer Graphics and Interactive Techniques, p.177-184. [doi:10.1145/133994.134036]

Lipman, Y., Sorkine, O., Cohen-Or, D., Levin, D., Rossi, C., Seidel, H.P., 2004. Differential Coordinates for Interactive Mesh Editing. Proc. 6th Int. Conf. on Shape Modeling and Applications, p.181-190. [doi:10.1109/SMI.2004.1314505]

Lorensen, W.E., Cline, H.E., 1987. Marching Cubes: a High Resolution 3D Surface Construction Algorithm. Proc. 14th Annual Conf. on Computer Graphics and Interactive Techniques, p.163-169. [doi:10.1145/37401.37422]

Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., Seidel, H.P., 2003. Multi-level partition of unity implicits. *ACM Trans. Graph.*, **22**(3):463-470. [doi:10.1145/882262.882293]

Praun, E., Finkelstein, A., Hoppe, H., 2000. Lapped Textures. Proc. 27th Annual Conf. on Computer Graphics and Interactive Techniques, p.465-470. [doi:10.1145/344779.344987]

Rohmer, D., Hahmann, S., Cani, M.P., 2009. Exact Volume Preserving Skinning with Shape Control. Proc. ACM SIGGRAPH/Eurographics Symp. on Computer Animation, p.83-92. [doi:10.1145/1599470.1599481]

Sederberg, T.W., Parry, S.R., 1986. Free-Form Deformation of Solid Geometric Models. Proc. 13th Annual Conf. on Computer Graphics and Interactive Techniques, p.151-160. [doi:10.1145/15922.15903]

Singh, K., Fiume, E., 1998. Wires: a Geometric Deformation Technique. Proc. 25th Annual Conf. on Computer Graphics and Interactive Techniques, p.405-414. [doi:10.1145/280814.280946]

Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C., Seidel, H.P., 2004. Laplacian Surface Editing. Proc. Eurographics ACM SIGGRAPH Symp. on Geometry Processing, p.175-184. [doi:10.1145/1057432.1057456]

Stam, J., 2003. Flows on surfaces of arbitrary topology. *ACM Trans. Graph.*, **22**(3):724-731. [doi:10.1145/882262.882338]

Theisel, H., Weinkauf, T., Hege, H.C., Seidel, H.P., 2005. Topological methods for 2D time-dependent vector fields based on stream lines and path lines. *IEEE Trans. Visual. Comput. Graph.*, **11**(4):383-394. [doi:10.1109/TVCG.2005.68]

van Wijk, J.J., 2003. Image Based Flow Visualization for Curved Surfaces. Proc. IEEE Visualization, p.123-130. [doi:10.1109/VISUAL.2003.1250363]

von Funck, W., Theisel, H., Seidel, H.P., 2006. Vector field based shape deformations. *ACM Trans. Graph.*, **25**(3):1118-1125. [doi:10.1145/1141911.1142002]

Zhang, E., Mischaikow, K., Turk, G., 2006. Vector field design on surfaces. *ACM Trans. Graph.*, **25**(4):1294-1326. [doi:10.1145/1183287.1183290]

Zhou, K., Huang, J., Snyder, J., Liu, X., Bao, H., Guo, B., Shum, H.Y., 2005. Large mesh deformation using the volumetric graph Laplacian. *ACM Trans. Graph.*, **24**(3):496-503. [doi:10.1145/1073204.1073219]