The 2015 World Congress on
Advances in Structural Engineering and Mechanics (ASEM15)
Incheon, Korea, August 25-29, 2015

Area-preserving Deformation of 3D Objects

Zhiwei Li[1)] and Kin-chuen Hui[2)]*

[1), 2)] *CAD Laboratory, Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong SAR, P. R. C.*
[2)] *kchui@mae.cuhk.edu.hk*

ABSTRACT

Modeling deforming shapes under constraint is a challenging task due to the need for maintaining a balance between the deformed shape quality and the required constraints. Recent works on area-preserving deformation mainly focus on the deformation of two dimensional shapes, while the area-preserving deformation of three dimensional shapes remains not well addressed. This paper describes an algorithm to deform a 3D object intuitively while maintaining its surface area. Unlike the classical approach, which always leads to non-linear and non-convex formulation which is computational expensive and may not be robust, we derived a simple three-step iterative optimization scheme to ensure a global optimum as the solution while preserving the surface area as desired. The result can be used in applications including the design of deformable objects and the estimation of the volume reduction of plastic waste.

1. INTRODUCTION

Applications of geometric modeling often involve physical simulation, such as engineering analysis or animation. CAD models constructed with geometric modeling have to be physically realizable and if the model is deformed, some properties, such as lengths, areas, or curvature may need to be preserved after the deformation. However, modeling deforming shapes under constraint is a complicated task due to the highly nontrivial interaction between the need for flexible local rules and suitable global constraints. One appropriate deforming algorithm should meet both the minimum smooth criteria and maximum preservation constraint.

Due to the wide range of applications in industrial and artistic design, there are large numbers of research work in constraint-based design. Positional constraints are used successfully in simple constrained deformations [Borrel 1994, Eigensatz 2009],

---

[1)] Graduate Student
[2)] Professor

multiresolution editing [Zorin 1997, Kobbelt 1998] and surface deformation based on differential coordinates [Sorkine 2004, Lipman 2004]. Volume constraints are well addressed in detail preserving deformation of B-spline surfaces [Sauvage 2008]. Differential normal or curvature constraints at fixed points in space are used to create a set of interpolating parameterized surface patches from scratch [Moreton 1992]. Volume-preserving algorithms are well addressed using FEM [Glowinski 1984], generalized direct manipulation FFD [Aubert 1997], discrete multilevel representations [Hirota 2000] and subspace gradient domain techniques [Huang 2006]. All these methods have merits simulating the deformation of complex solids or shell elements.

However, in real world, there are a great collection of articles whose deformation involves the surface area to be nearly constant while the volume decreases. One can easily name examples such as folding clothes with material of little elasticity, or twisting a plastic bottle or pressing a cushion covered with silk material. Unfortunately, despite the need, little research has been conducted in this area. In this paper, we proposed an area-preserving deformation (APD) algorithm for deforming a 3D object while maintaining its total area as far as possible.

Usually, the main ingredient of constrained deformation algorithm is a global quadratic minimization problem [Botsch 2008, Wardetzky 2007,], whose solution, given certain constraints either derived from mathematical equation or user interaction, is the desired deformed surface. Minimization is achieved by solving a non-linear function. However, this often leads to expensive computation and the optimization algorithm is highly restricted by the convexity of both the target function and the constraint. If either is non-convex, it will be hard to find the global optimum as a stable solution. Therefore, we derive a simple and efficient discrete method which preserves the area as much as possible.

From the classical shell energy model, as the energy function measures the difference between the deformed and undeformed surface, we can reach the conclusion that the energy is minimized when the transformation between the two surfaces is rigid. In this case, surface area is fully preserved. However, in real deformation scenarios, local rigidity cannot hold. Thus it is useful to preserve the area while having a deformation that is as-rigid-as-possible.

Our framework uses a simple three-step algorithm that ensures a global optimum and is easy to implement. In the first step, which we call “pre-computation”, we use the As-rigid-as-possible (ARAP) method proposed by Sorkine [Sorkine 2007] to give a rough estimation of the deformed surface and compute the change in surface area. In the second step, which we call “correction”, according to the area change, we give each vertex on the mesh a correcting vector to compensate the change. In the third step, we use least-square fitting to form a smooth surface which better preserves the area.

2. AREA-PRESERVIND DEFORMATION

2.1 Deformable Surface Setup and Algorithm Framework

Let S denote a deformable surface by a discrete triangle mesh $\mathbf{M}=(\mathbf{V},\ \mathbf{E},\ \mathbf{T})$, whose topology is determined by $\mathbf{V}=\{\mathbf{v}_i\}$ the set of n vertices, $\mathbf{E}\{\mathbf{e}_{ij}:=\mathbf{v}_i-\mathbf{v}_j\}$ the set of m edges and $\mathbf{T}=\{\mathbf{t}_{ijk}\}$ the triangle face set with $1\le i,\ \ j,\ \ k\le n$. Mesh $\mathbf{M}$ is defined by the vertex position $\mathbf{v}_i$. We also define the deformed surface of S as S', which differs geometrically but has the same connectivity.

2.2 *Algorithm Framework*

2.2.1 *As-rigid-as-possible deformation algorithm*

The basic idea of ARAP is quite simple. It assumes that each triangle on the mesh is rigid and there is only rotational transformation without stretching or scaling. Thus there exists a rotation matrix which can deform S_0 into S. Energy function can be formulated as below

$$E_{ARAP}=\sum_{i=1}^{n}\sum_{j\in N(i)} w_{i,j}\left\|\left(\mathbf{v}_i-\mathbf{v}_j\right)-\mathbf{R}_i(\mathbf{v}_i{}^0-\mathbf{v}_j{}^0)\right\|^2 \tag{1}$$

where $\mathbf{v}_j$ is the one-ring neighbor of $\mathbf{v}_i$, $\mathbf{R}_i$ denotes a rotational matrix for each triangle and $w_{i,j}$ stands for the weight of edge vector $\mathbf{v}_i-\mathbf{v}_j$ with cotangent weight proposed by Pinkall et al.[Pinkall and Polthier 1993].

The ARAP energy is minimized by an iterative process of rotating the triangles and generating a smooth surface from a least-square solution.

2.2.2 *The optimization scheme can be described in 3 steps*

i. *Estimation and Pre-computing* ($S_0\to S$)

We denote the original surface as S_0 and minimize the ARAP energy to get a rough estimation of the deformed surface and keep the record of the corresponding vertex position. The ARAP deformation is triggered by fixing certain vertices as geometric constraint while moving some others directly to a different location. The surface areas before and after the deformation are recorded as A_0 and A, which are computed by summing every triangles' area (See Fig. 1). Then the area change can be expressed as

$$\Delta A=\sqrt{\frac{A_0}{A}} \tag{2}$$

We define three types of vertices after an ARAP deformation as listed below.

a. Base $\mathbf{B}_i(b_{ix},b_{iy},b_{iz})$ - the fixed vertices

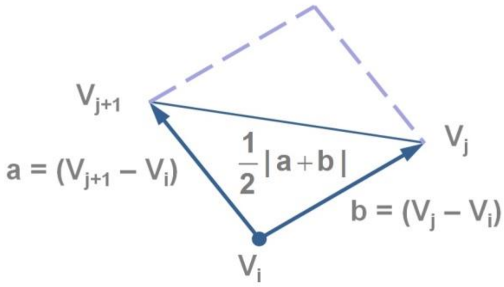

Fig. 1 Area of a triangle expressed as half the magnitude of the cross product of two adjacent edge vectors.

b. Handle $\mathbf{H}_i(h_{ix}, h_{iy}, h_{iz})$ - points which are used for direct manipulation and which trigger the deformation
c. Free $\mathbf{F}_i(f_{ix}, f_{iy}, f_{iz})$ - points that belong to neither base nor handle

If the base contains l vertices and the handle contains k vertices, we can simplify the base and handle to two points $\mathbf{B}_c$ and $\mathbf{H}_c$ can be calculated as

$$\mathbf{B}_c = \frac{1}{l}\sum_{i=1}^{l}\mathbf{B}_i, \quad 1 \le i \le l \tag{3}$$

$$\mathbf{H}_c = \frac{1}{k}\sum_{i=1}^{k}\mathbf{H}_i, \quad 1 \le i \le k \tag{4}$$

ii. *Correction* ($S \rightarrow S'$)

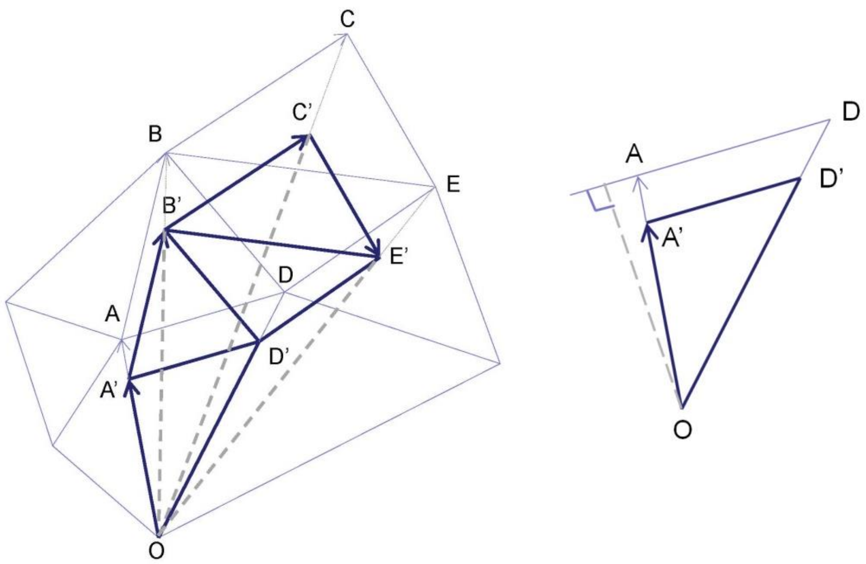

Fig. 2 A sketch of part of a triangle mesh before and after correction

Fig. 2 shows a part of the triangle mesh, lines in light blue represent the estimated mesh after applying ARAP, while lines in dark blue shows its corrected version. In order to make correction of the area change, the relation between the area and each triangle edge should be studied. Let us take triangle OAD and its corrected counterpart $OA'D'$ as example. If both OA' and OD' changes proportionally to its original length and goes along the same edge vector. The area change is proportion to the squares of the edge length change. Moreover, take $OABC$ for instance, assume O to be the fixed reference point, when A changes to A' and B to B', then C to C', as it is a scaling transformation, the changing trajectory of C still goes along the original vector from O to C. Therefore, we reach the conclusion that the correction can be made by compensating the length change in each triangle edge, along the vector from the fixed reference point in space to each vertex. Details are as below.

a. The moving vector $\mathbf{d}_{\mathbf{F}_i}$ for type "Free" vertices during correction can be expressed as

$$\mathbf{d}_{\mathbf{F}_i} = \mathbf{F}_i - \mathbf{B}_c \tag{5}$$

One can therefore obtain the vertex position $\mathbf{F}_i'$ for type "Free" after correction as

$$\mathbf{F}_i' = \mathbf{B}_c + \Delta A \cdot \mathbf{d}_{\mathbf{F}_i} = \mathbf{B}_c + \Delta A \cdot (\mathbf{F}_i - \mathbf{B}_c) \tag{6}$$

b. The difference vector $\mathbf{d}_{\mathbf{H}_c}$ between Handle and Base can be expressed as

$$\mathbf{d}_{\mathbf{H}_c} = \mathbf{H}_c - \mathbf{B}_c \tag{7}$$

while the vertex position H_i' for type "Handle" after correction is

$$\mathbf{H}_i' = \mathbf{H}_i + (\Delta A - 1) \cdot \mathbf{d}_{\mathbf{H}_c} = \mathbf{H}_i + (\Delta A - 1) \cdot (\mathbf{H}_c - \mathbf{B}_c) \tag{8}$$

c. The Base remains unaltered.

iii. *Blending*

The final deformed vertex position V_i' contains three categories, F_i', H_i' and B_i'. A smooth and area-preserving surface can then be obtained by minimizing the function (9).

$$\sum_{i=1}^{n} (\mathbf{v}_i' - \mathbf{v}_i)^2, \quad 1 \le i \le n \tag{9}$$

Then we calculate the current area A' and go to step one again, evaluating $\sqrt{\frac{A_0}{A'}}$. The iteration stops until we get the area change within a desired value.

3. IMPLEMENTATION RESULTS

3.1 Twisting Operation

In a twisting operation, the top of the bottle functions as the handle. The bottle is being twisted 60 and 120 degree respectively. Vertices at the bottom of the bottle are the fixed base.

Fig. 3 shows the result of the twisting operation. (a) is the original bottle model, (b) and (c) shows the corresponding result after twisting with the handle for 60 and 120 degree's twisting of the handle.

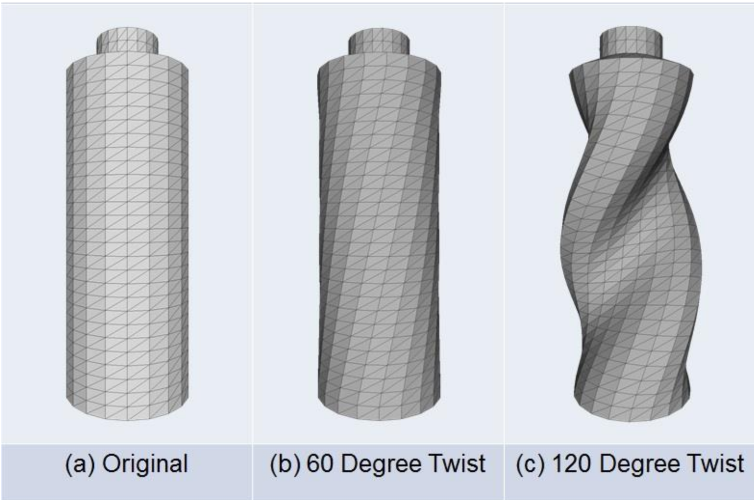

Fig. 3 Result of twisting operation

Table 1 gives details of the surface area after ARAP and our area-preserving deformation algorithm respectively, with the area change tolerance of 0.1%. It is clear that APD (Area Preserved Deformation) preserves the area effectively.

Table 1. Twisting results of ARAP deformation and area-preserving deformation.

	Original	60 Degree Twist	120 Degree Twist
ARAP Area	79.305	77.781	78.957
Area Change	—	↓ 1.92%	↓ 0.44%
APD Area	79.305	79.231	79.267
Area Change	—	↓ 0.09%	↓ 0.047%

3.2 *Bending Operation*

Bending operation is performed by pressing the tail of a duck model. The beak of the duck is chosen as the fixed base, while the handle locates at the tail, along the intersection between the model and the reference plane.

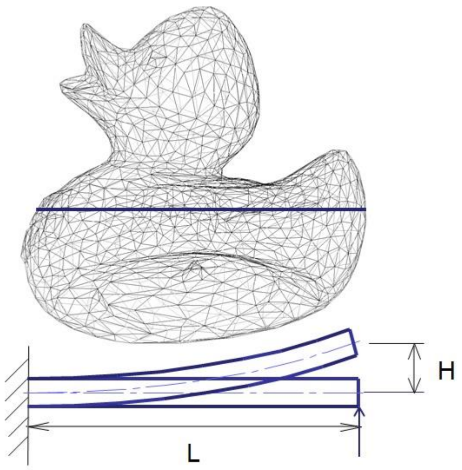

Fig. 4

As in Fig. 4, L is the length of the reference plane intersects the duck model, H measures the bending distance of such plane. Fig. 5 shows the results of two bending operations.

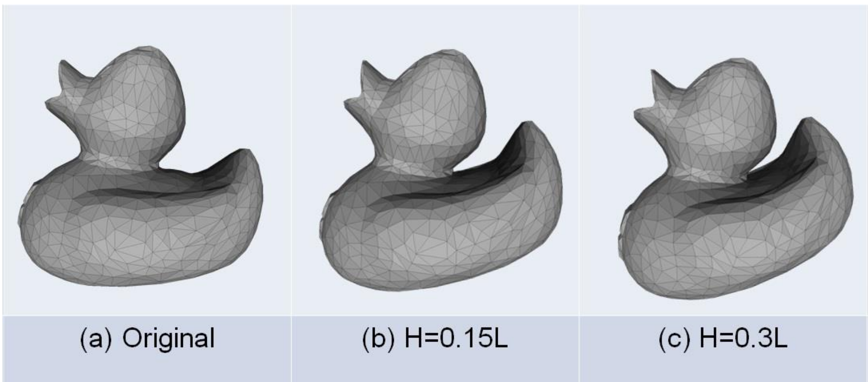

Fig. 5 Result of bending operation

Table 2 gives the details of the surface area in bending operation using ARAP and our area-preserving deformation algorithm, with the area change tolerance of 0.1%.

Table 2. Bending results of ARAP deformation and area-preserving deformation.

	Original	0.15L Bend	0.3L Bend
ARAP Area	3725	3699.5	3674.8
Area Change	—	↓ 0.68%	↓ 1.35%
APD Area	3725	3721.6	3722.8
Area Change	—	↓ 0.091%	↓ 0.059%

4. CONCLUSION AND DISCUSSION

We derived a simple and robust 3-step iterative algorithm to preserve the surface area of 3D models under deformation. Two different deformation – twisting and bending are performed on two different models. From the experimental data, it is clear that our APD algorithm works effectively. By compensating the change in the as-rigid-as-possible deformation and applying suitable constraints, the area is controlled within the threshold value.

This algorithm has a wide range of potential applications including the design of deformable objects, simulation of inelastic fabric and the estimation of the volume reduction of plastic waste.

ACKNOWLEDGEMENT

This work is partially supported by a Grant from the Research Grants Council of the Hong Kong Special Administration Region (Project No. 412913).

REFERENCES

Aubert, F. and Bechmann, D. (1997), “Volume-preserving space deformation”, *Computer & Graphics,* **21**(5), pp. 625-639.

Borrel, P. and Rappoport, A. (1994), “Simple Constrained Deformations for Geometric Modeling and Interactive Design”, *ACM Transactions on Graphics*, **13**(2), 137-155.

Botsch, M. and Sorkine, O. (2008), “On linear variational surface deformation methods”, *Visualization and Computer Graphics,* **14**(1), 213-230.

Eigensatz, M.andPauly, M (2009), “Positional, Metric, and Curvature Control for Constraint-Based Surface Deformation”, *Computer Graphics Forum*, **28**(2), 551-558.

Glowinski, R. and Tallec, P.L. (1984), “Numerical solution of problems in incompressible finite elasticity by augmented Lagrangian methods II. Three-dimesional problems”. *SIAM Journal on Applied Mathematics,* **44**(4), pp. 710-733.

Hirota, G., Maheshwari, R. and Lin M.C. (2000), “Fast volume-preserving free-form deformation using multi-level optimization”, *Computer-Aided Design*, **32**(8-9):499-512.

Huang, J., Shi, X., Liu, X., Zhou, K., Wei, L., Teng, S., Bao, H., Guo, B. and Shum, H.-Y. (2006), “Subspace gradient domain mesh deformation”, *ACM SIGGRAPH 2006 Papers , ACM,* 1126-1134.

Kobbelt, L., Campagna, S., Vorsatz, J. and Seidel, H.-P. (1998), “Interactive multi-resolution modeling on arbitrary meshes”, *Proceedings of the 25th annual conference on Computer graphics and interactive techniques, ACM,* 105-114.

Lipman Y., Sorkine O., Cohen-Or, D., Levin, D., Rössl, C. and Seidel, H.-P. (2004), “Differential coordinates for interactivemesh editing”, *Proceedings of the Shape Modeling International 2004, IEEE Computer Society,* 181-190.

Moreton, H.P. and Séquin, C.H. (1992), "Functional optimization for fair surface design", *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, 167-176.

Sauvage, B., Hahmann, S., Bonneau, G.-P., Elber, G. (2008), "Detail preserving deformation of B-spline surfaces with volume constraint", *Computer Aided Geometric Design*, **25**(8), 678–696.

Sorkine, O. and Alex, M. (2007), "As-rigid-as-possible surface modeling", *Proceedings of the fifth Eurographics symposium on Geometry processing,* 109-116.

Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C. and Seidel, H.-P. (2004), "Laplacian surface editing", *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, ACM,* 175-184.

Wardetzky, M., Bergou, M., Harmon, D.,Zorin, D. and Grinspun, E. (2007), "Discrete Quadratic Curvature Energies", *Computer Aided Geometric Design,*Vol. **24**(8-9), 499–518.

Zorin, D., Schröder, P. and Sweldens, W. (1997), "Interactive multiresolution mesh editing", *Proceedings of the 24th annual conference on Computer graphics and interactive techniques,* 259-268.