

Cute Balloons with Thickness

Qingyun Wang¹, Xuehui Liu¹(✉), Xin Lu¹, Jianwen Cao¹, and Wen Tang²

¹ State Key Laboratory of Computer Science, Institute of Software,
China Academy of Sciences, Beijing, China
{wangqy,lxh,lxin}@ios.ac.cn, jianwen@iscas.ac.cn

² The Faculty of Science and Technology, Bournemouth University, Poole, UK
wtang@bournemouth.ac.uk

Abstract. Based on the finite element method, we present a simple volume-preserved thin shell deformation algorithm to simulate the process of inflating a balloon. Different from other thin shells, the material of balloons has special features: large stretch, small bend and shear, and incompressibility. Previous deformation methods often focus on typical three-dimensional models or thin plate models such as cloth. The rest thin shell methods are complex or ignore the special features of thin shells especially balloons. We modify the triangle element to simple three-prism element, ignore bending and shearing deformation, and use volume preservation algorithm to match the incompressibility of balloons. Simple gas model is used, which interacts with shells to make the balloons inflated. Different balloon examples have been tested in our experiments and the results are compared with those of other methods. The experiments show that our algorithm is simple and effective.

Keywords: Finite element method · Physically based simulation · Thin shell · Volume preservation · Deformation animation of balloons

1 Introduction

Physically based deformation simulation has emerged in the late 1980s to make animations more physically plausible and to make the simulation of complex passively moving object easier. The simulation of thin shells is one of the difficult issues among the physically based deformation animation field. Rubber balloons belong to thin shells with large deformation.

Similar to thin plates, thin shells are thin flexible structures with a thickness much more smaller than other dimensions. The difference between them is that thin shells have a curved undeformed configuration (e.g., leaves, hats, balloons) while thin plates have a flat one (e.g., clothes). Thin shells are difficult to simulate because of the degeneracy of one dimension. They should not be treated as three-dimensional solids, otherwise the numerics become ill-conditioned. Due to the similarity of thin shells and thin plates, thin shells are treated as curved surface in many works [1, 18] just like the approximation made for thin plates. Unfortunately, the thickness, which is very important for some materials such as rubber,

is discarded in such methods. Different thickness will lead to different elastic force. Ignoring the thickness will cause artifacts. Additionally, for some complex object, the material points demanded by these works [1, 18] are hard to get.

Compared to typical thin shells, the material of balloons has special features [1]. First, the main force leading to deformation is stretching force, while bending force is tiny. Second, significant transverse shearing should not exist. Last, the deformation of balloons is volume-preserving. A balloon becomes thinner when inflated. By adding gas, the expansion of the balloon under gas pressure will be stretched with the elastic force increasing. Meanwhile, the volume of gas is increasing and the pressure is dropping. In the end, gas pressure and elastic force balance.

Considering the characteristics of thin shells and balloons, we propose a simple volume-preserved thin shell algorithm to simulate the process of inflating a balloon, based on finite element method.

Our algorithms are mainly divided into two phases. In the first phase, finite element method is applied to solve the temporary shape of the object in the next time step. In the second phase, volume-preservation constraint is used to get the final shape. In contrast to other system, we adopt three-prism element instead of triangle element in order to involve thickness. The three-prism element is modified from triangle element by adding a thickness property. Despite there are six vertices in a three-prism, we need not to use all of them as there are not significant transverse shearing in rubber material. In the first phase, our method is similar to the finite element method based on triangle element. The thickness is treated as a parameter. The force caused by thickness is counted as the force of vertices. Thus, it is necessary to update the thickness and vertices in volume-preservation algorithm. A position based method [28] is applied in our volume-preservation algorithm, which is simple and effective. The gas pressure of every point inside the balloon is equal and the coupling of gas and balloon is simplified.

Our contributions:

- A simple thin shell finite element method with modification from finite element method based on triangle element.
- A volume constraint for thin shells.
- A method simulating the process of inflating a balloon with a simple gas model and capturing thickness effect.

2 Related Work

Physically based deformation simulation has emerged in the late 1980s. One of the first studies was the pioneering work of Terzopoulos et al. [2], which presented a simulation method about deformable objects based on finite differences. After that, many different approaches were proposed, such as mass-spring system [3, 4], the boundary element method [5], the finite element method [6]. More approaches can be found in surveys of this field [7, 8].

Due to the limitation of the calculation speed of computers, most of early methods were based on linear elasticity, which will lead to artifact in large deformation. Non-linear finite element methods were introduced to improve the accuracy of simulation [9, 10], which cost too much computation time. Considering

the fact that large deformation is caused by rotation in many deformable objects, corotated finite element method [11, 12] was presented to simulation such objects with acceptable speed and accuracy. However, such methods are not suitable for objects with large stretching deformation. McAdams et al. [13] overcame the shortcomings of corotated method to simulate the deformation of skin based on hexahedral element. But the improvement could not be extended to other types of element such as triangle element.

Thin Shells. Due to the degeneracy of one dimension (the high ratio of width to thickness), thin shells are difficult to simulate. Robust finite element based thin shell simulation is an active and challenging research area in physics, CAD and computer graphics. Arnold [14] analyzed thin shells and pointed out the issues in simulation. Based on Kirchhoff-Love thin shell theory, Cirak et al. [15] adopted subdivision basis as the shape function in finite element method. The accuracy of simulation was improved, neglecting the low speed. Green et al. [16] applied the method of Cirak et al. [15] in computer graphics and improved the speed with multi-level method. Grinspun et al. [17] also proposed an accelerated framework, which was simpler and more adaptable. Grinspun et al. [18] presented the simplest discrete shells model to simulate thin shells with large bending deformation such as hats and papers. Bonet et al. [1] simulated thin shells under gas pressure with the approach for clothes. 2D material points are demanded in this method, which are difficult to calculate in some complex thin shells. Inspired by the work of Bonet et al. [1], Skouras et al. [19] modified the form of deformation gradient to make every element volume-preserved during deformation and got rid of the demand of material points. Nevertheless, the new deformation gradient will lead to a complex system, which may be ill-conditioned. Most of the works above use triangle element, ignoring the thickness when simulating thin shells. But for materials like rubber, thickness is a very important property, which should be considered. On the other hand, thin shells should not be simulated based on tetrahedral element, as the degeneracy of one dimension. In this paper, we present a compromise method by using three-prism elements, which is simple and keep the thickness information meanwhile.

Gas Model. Gas is a type of fluid, which is a active research area in recent years. Guendelman et al. [20] and Robinson et al. [21] coupled fluid with thin shells and thin plates. The coupling process is very complex as thin structures and fluid belong to two different system. Batty et al. [22] accelerated the coupling of fluid and rigid bodies. However, the coupling of fluid and soft bodies is still a difficult problem. It is not necessary to use such complex system, when there is not significant movement of gas. Chen et al. [23] proposed a three-layer model to simulate the air effect between clothes and soft bodies. The air flow was calculated by simple diffusion equation. In the deformation of balloons, the equilibrium of gas pressure force and elastic force is achieved quickly so we can just ignore the movement and treat the gas uniform [19].

Volume-Preservation. Volume loss can be an obvious artifact when simulating large deformation. With meshless method, Müller et al. [24] adjusted material parameters such as Poisson’s ratio to make the deformation volume-preserved. Bargeil et al. [25] proposed a plasticity model to preserve volume, which could only be used in offline simulation. Local volume-preserving method was presented by Irving et al. [26], which preserve the local volume around every vertex. Position based method could also be used to preserve volume by adding volume constraint [28]. Müller et al. [29] then improved this method with multi-grid to accelerate the solving of constraints. Similar to Müller [28], Diziol et al. [30] also use position based method, but they solve constraints of not only positions but also velocities.

3 Overview

The balloon model is expressed by a triangles mesh, with gas inside. The mesh is represented by three-prism elements with a set of N vertices. A vertex $i \in [1, \dots, N]$ has a mass m_i , a initial position \mathbf{x}_i^0 , a current position \mathbf{x}_i^n and a current velocity \mathbf{v}_i^n . With a time step Δt , the whole volume-preserved deformation algorithm for thin shells is as follows:

ALGORITHM 1. BALLOON INFLATED DEFORMATION.

```

1  Set parameters: gas pressure  $p$ , balloon density  $\rho$ , initial thickness  $h^e$ .
2  for all vertices  $i$ 
3      do initialize  $\mathbf{x}_i^0, \mathbf{v}_i^0$ , compute mass  $m_i$ 
4  while system dose not converge
5      do for all elements  $e$ 
6          do compute  $\mathbf{f}_i^{elastic}, \mathbf{f}_i^p$  of the three vertices
7              compute  $\delta \mathbf{f}^{elastic} = -\mathbf{K}^{elastic} \delta \mathbf{x}, \delta \mathbf{f}^p = -\mathbf{K}^p \delta \mathbf{x}$ 
8              compute the time integration by implicit method
9              solve the equation to get all  $\mathbf{v}_i^{n+1}, \mathbf{x}_i^{n+1}$ 
10             Volume preserving, update all  $\mathbf{x}_i^{n+1}, h^e$ 
11             for all vertices  $i$ 
12                 do  $\mathbf{v}_i^{n+1} = (\mathbf{x}_i^{n+1} - \mathbf{x}_i^n) / \Delta t$ 

```

Lines 1–3 set the parameters and initialize the positions and velocities of vertices. The core of the algorithm are two parts, the finite element method part (lines 5–9) and volume preserving part (line 10). Lines 11–12 update the positions and velocities at last.

In the finite element method part, the elasticity forces and the pressure forces of vertices are computed according to current positions \mathbf{x}_i^n and initial positions \mathbf{x}_i^0 in every element in line 6. In line 7, the variations related to stiffness matrix and gas tangent matrix of every element are computed, which will be assembled in the variations related to stiffness matrix and gas tangent matrix of the system. After these computed for all elements, the implicit time integration is used to construct system equation in line 8 by combining the matrixes above. At last, the

new velocities are solved by Newton-Raphson method and Conjugate Gradient method in line 9.

In volume preserving part, current positions \mathbf{x}_i^{n+1} and thicknesses h^e are used to compute the change of volume, and the positions and thicknesses are updated in order to preserve the volume.

4 Balloon Model

In most of previous works, thin shells are simulated by triangle elements, which ignore the thickness information, or tetrahedral elements, which may be ill-conditioned. We present a three-prism element to keep the thickness. As there is no significant shear in rubber material, we just add a thickness property on the typical triangle element to get a three-prism element (see Fig. 1).

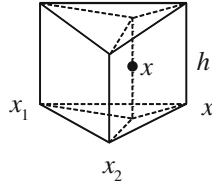


Fig. 1. Interpolating method of three-prism element.

The deformation of objects is represented by the deformation function $\phi : \Omega \rightarrow \mathbf{R}^3$, which maps a material point $\bar{\mathbf{x}}$ to its deformed point $\mathbf{x} = \phi(\bar{\mathbf{x}})$. Inside the three-prism element, the location of a point $\mathbf{x}(\alpha)$ can be interpolated by three vertices and the thickness property (see Fig. 1) and α is the interpolating parameter.

In element e , let $\bar{\mathbf{x}}_1^e, \bar{\mathbf{x}}_2^e, \bar{\mathbf{x}}_3^e \in \mathbf{R}^3$ denote the vertices positions in undeformed configuration, $\bar{h}^e \in \mathbf{R}$ denote the thickness, and $\bar{\mathbf{e}}_{ij} = \bar{\mathbf{x}}_j^e - \bar{\mathbf{x}}_i^e$ denote the edges. Let $\mathbf{x}_1^e, \mathbf{x}_2^e, \mathbf{x}_3^e \in \mathbf{R}^3$ denote the respective deformed vertices positions, $h^e \in \mathbf{R}$ denote the deformed thickness and $\mathbf{e}_{ij} = \mathbf{x}_j^e - \mathbf{x}_i^e$ denote the deformed edges. Using linear interpolating method, a point within element e in undeformed configuration $\bar{\mathbf{x}}^e(\alpha)$ and its respective deformed position $\mathbf{x}^e(\alpha)$ can be represented as the same form

$$\bar{\mathbf{x}}^e(\alpha) = (1 - \alpha_1 - \alpha_2)\bar{\mathbf{x}}_1^e + \alpha_1\bar{\mathbf{x}}_2^e + \alpha_2\bar{\mathbf{x}}_3^e + \alpha_3\bar{h}^e\bar{\mathbf{d}}, \bar{\mathbf{d}} = \frac{\bar{\mathbf{e}}_{12} \times \bar{\mathbf{e}}_{13}}{|\bar{\mathbf{e}}_{12} \times \bar{\mathbf{e}}_{13}|} \quad (1)$$

Within every element, the deformation is assumed to be constant, and can be described by the deformation gradient $\mathbf{F}^e \in \mathbf{R}^{3 \times 3}$

$$\mathbf{F}^e = \frac{\partial \mathbf{x}^e(\alpha)}{\partial \bar{\mathbf{x}}^e(\alpha)} = \frac{\partial \mathbf{x}^e(\alpha)}{\partial \alpha} \cdot \left(\frac{\partial \bar{\mathbf{x}}^e(\alpha)}{\partial \alpha} \right)^{-1} \quad (2)$$

From Eq. (1), the partial derivative of $\bar{\mathbf{x}}^e(\alpha)$ to α is

$$\frac{\partial \bar{\mathbf{x}}^e(\alpha)}{\partial \alpha} = [\mathbf{e}_{12} \ \mathbf{e}_{13} \ \mathbf{d}h^e], \quad \frac{\partial \bar{\mathbf{x}}^e(\alpha)}{\partial \alpha} = [\bar{\mathbf{e}}_{12} \ \bar{\mathbf{e}}_{13} \ \bar{\mathbf{d}}\bar{h}^e] \quad (3)$$

At last, the deformation gradient \mathbf{F}^e can be represented as

$$\mathbf{F}^e = [\mathbf{e}_{12} \ \mathbf{e}_{13} \ \mathbf{d}h^e] \cdot [\bar{\mathbf{e}}_{12} \ \bar{\mathbf{e}}_{13} \ \bar{\mathbf{d}}\bar{h}^e]^{-1} \quad (4)$$

Deformation could cause strain energy, which is defined by deformation gradient

$$E = \int_{\Omega} \psi(\mathbf{F}) d\bar{\mathbf{x}} \quad (5)$$

where ψ is the energy density function as a function of deformation gradient to measure the strain energy per unit undeformed volume. Different material has different energy density. Take St.Venant-Kirchhoff material for example

$$\psi(\mathbf{F}) = \mu \mathbf{E} : \mathbf{E} + \frac{\lambda}{2} tr^2(\mathbf{E}) \quad (6)$$

where μ, λ are Lamé coefficients of the material and $\mathbf{E} \in \mathbf{R}^{3 \times 3}$ is Green strain tensor, which is a nonlinear function of deformation gradient

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^T \mathbf{F} - \mathbf{I}) \quad (7)$$

With strain energy, the elastic forces are defined as

$$\begin{aligned} \mathbf{f}_i^{elastic} &= -\frac{\partial E}{\partial \mathbf{x}_i} = \sum_e \left(-\frac{\partial E^e}{\partial \mathbf{x}_i} \right) = \sum_e \mathbf{f}_i^e \\ &= \sum_e -V^e \frac{\partial \psi(\mathbf{F}^e)}{\partial \mathbf{x}_i} = \sum_e -V^e \frac{\partial \psi(\mathbf{F}^e)}{\partial \mathbf{F}^e} \frac{\partial \mathbf{F}^e}{\partial \mathbf{x}_i} \\ &= \sum_e -V^e \mathbf{P}(\mathbf{F}^e) \frac{\partial \mathbf{F}^e}{\partial \mathbf{x}_i} \end{aligned} \quad (8)$$

where V^e is the volume of element e , $\mathbf{P}(\mathbf{F}) = \frac{\partial \psi}{\partial \mathbf{F}}$ is 1st Piola-kirchhoff stress tensor, which of St.Venant-Kirchhoff material is

$$\mathbf{P}(\mathbf{F}) = \mathbf{F}[2\mu \mathbf{E} + \lambda tr(\mathbf{E})\mathbf{I}] \quad (9)$$

If we use explicit time integration, we can compute the accelerations of all vertices and then update the positions and velocities. However, the explicit time integration is not stable in large time step. In order to apply implicit time integration, the stiffness matrix need to be computed

$$\mathbf{K}^{elastic} = -\frac{\partial \mathbf{f}^{elastic}}{\partial \mathbf{x}} \quad (10)$$

From Eqs. (4) and (8), we can see that deformation gradient and stress tensor are complex functions of \mathbf{x} . Thus solving $\mathbf{K}^{elastic}$ directly is difficult. However,

$\mathbf{K}^{elastic}$ always multiply a vector like $\mathbf{K}^{elastic}\mathbf{w}$, so we just compute the variation of elastic forces $\delta\mathbf{f}^{elastic} = -\mathbf{K}^{elastic}\delta\mathbf{x}$

$$\delta\mathbf{f}_i^{elastic} = \sum_e -V^e \mathbf{P}(\mathbf{F}^e; \delta\mathbf{F}^e) \frac{\partial \delta\mathbf{F}^e}{\partial \delta\mathbf{x}_i} \quad (11)$$

Now we can use implicit time integration to solve positions and velocities.

5 Gas Model

As the equilibrium of gas pressure force and elastic force is achieved quickly when inflating the balloon, we assume that the gas pressure is uniform and the gas pressure of every point inside the balloon is the same. The pressure p and the volume of gas V have a relationship as

$$pV = nRT \quad (12)$$

where n is the amount of gas contained in volume V , T is the temperature and R is the gas constant. In an enclosed balloon, the amount of gas, the temperature and the gas constant do not change, so the product of pressure and volume remains invariant. Increasing the volume will decrease the pressure.

In our model, when the mesh is very fine, the volume of the gas can be approximately formed from tetrahedrons made up of triangles and the origin point (see Fig. 2)

$$V = \sum_e \frac{1}{3} \cdot \frac{1}{3} (\mathbf{x}_1^e + \mathbf{x}_2^e + \mathbf{x}_3^e) \cdot \mathbf{n}^e \cdot A^e \quad (13)$$

where \mathbf{n}^e is the normal vector of element e and A^e is the area.

With the initial pressure p_0 , initial volume V_0 and current volume V , current pressure is defined as

$$p = \frac{p_0 V_0}{V} \quad (14)$$

The value of the pressure force of element e is the product of the pressure p and the triangle area A^e of element e . The direction of the pressure force of element e is the normal vector of the triangle \mathbf{n}^e . The discrete nodal pressure forces can be derived as

$$\mathbf{f}_i^p = \sum_{e \in T^i} \frac{1}{3} p A^e \mathbf{n}^e = \sum_{e \in T^i} \frac{1}{3} p \cdot \frac{1}{2} (\mathbf{e}_{12}^e \times \mathbf{e}_{13}^e) \quad (15)$$

where T^i is the set of elements incident to vertex i , $\frac{1}{3}$ are weights.

The tangent matrix of pressure force is calculated and we neglect the derivative of area and normal vector to vertex position in order to make tangent matrix positive definite. The tangent matrix between vertex i, j is derived as

$$\begin{aligned} [\mathbf{K}^p]_{ij} &= \sum_{e \in T^i} -\frac{\partial [\mathbf{f}_i^p]^e}{\partial \mathbf{x}_j^T} \\ &\approx -\sum_{e \in T^i} \frac{1}{3} \left(\frac{\partial p}{\partial \mathbf{x}_j^T} A^e \mathbf{n}^e \right) \\ &\approx \frac{1}{27} \frac{p_0 V_0}{V^2} \left(\sum_{e \in T^i} A^e \mathbf{n}^e \right) \left(\sum_{e \in T^j} A^e \mathbf{n}^e \right)^T \end{aligned} \quad (16)$$

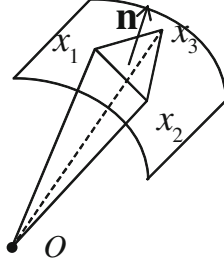


Fig. 2. Volume of the gas.

The tangent matrix is dense while the stiffness matrix is sparse. So the time complexity of the tangent matrix is $O(N^2)$. In order to speed up, just like the stiffness matrix, we compute the variation of pressure forces $\delta \mathbf{f}^p = -\mathbf{K}^p \delta \mathbf{x}$ directly. Let $\mathbf{A}\mathbf{n}_i = \sum_{e \in T^i} A^e \mathbf{n}^e$, the variation of pressure forces is derived as

$$\delta \mathbf{f}_i^p = - \sum_j [\mathbf{K}^p]_{ij} \delta \mathbf{x}_j = - \frac{1}{27} \frac{p_0 V_0}{V^2} \mathbf{A}\mathbf{n}_i \left(\sum_j (\mathbf{A}\mathbf{n}_j)^T \delta \mathbf{x}_j \right) \quad (17)$$

The time complexity of above is $O(N)$.

6 Time Integration

Implicit time integration is used as in [31] and we get

$$\mathbf{M}\mathbf{a} = \mathbf{f}^{elastic}(\mathbf{x}^{n+1}) + \mathbf{f}^p(\mathbf{x}^{n+1}) - \mathbf{C}\mathbf{v}^{n+1} + \mathbf{f}^{ext} \quad (18)$$

where $\mathbf{a} = \frac{\mathbf{v}^{n+1} - \mathbf{v}^n}{\Delta t}$ is the acceleration of vertices, \mathbf{v}^n is the velocity matrix at time n , \mathbf{C} is the damping matrix, \mathbf{f}^{ext} is the external forces such as gravity and \mathbf{M} is the mass matrix. The discrete nodal mass is derived as

$$m_i = \sum_{e \in T^i} w_i \rho \cdot V^e \quad (19)$$

where w_i are weights and ρ is the density of material.

As Eq. 18 is nonlinear, we use Newton-Raphson method to solve it. We will construct sequences of approximations $\mathbf{x}_{(k)}^{n+1} : \mathbf{x}_{(0)}^{n+1}, \mathbf{x}_{(1)}^{n+1}, \mathbf{x}_{(2)}^{n+1}, \dots$ and $\mathbf{v}_{(k)}^{n+1} : \mathbf{v}_{(0)}^{n+1}, \mathbf{v}_{(1)}^{n+1}, \mathbf{v}_{(2)}^{n+1}, \dots$ such that $\lim_{k \rightarrow \infty} \mathbf{x}_{(k)}^{n+1} = \mathbf{x}^{n+1}$ $\lim_{k \rightarrow \infty} \mathbf{v}_{(k)}^{n+1} = \mathbf{v}^{n+1}$. Initialize $\mathbf{x}_{(0)}^{n+1} = \mathbf{x}^n, \mathbf{v}_{(0)}^{n+1} = \mathbf{v}^n$ and define the correction variables

$$\Delta \mathbf{x}_{(k)} = \mathbf{x}_{(k+1)}^{n+1} - \mathbf{x}_{(k)}^{n+1}, \Delta \mathbf{v}_{(k)} = \mathbf{v}_{(k+1)}^{n+1} - \mathbf{v}_{(k)}^{n+1} \quad (20)$$

Linearize the elastic force and pressure force as

$$\mathbf{f}(\mathbf{x}^{n+1}) \approx \mathbf{f}(\mathbf{x}^n) + \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Delta \mathbf{x} \quad (21)$$

Then we use Eqs. 18, 20 and 21 to derive system linear equation

$$\begin{aligned} \frac{\mathbf{M} \left(\mathbf{v}_{(k+1)}^{n+1} - \mathbf{v}^n \right)}{\Delta t} &= \mathbf{f}^{elastic} \left(\mathbf{x}_{(k)}^{n+1} \right) + \mathbf{f}^p \left(\mathbf{x}_{(k)}^{n+1} \right) + \mathbf{f}^{ext} \\ -\mathbf{K}^{elastic} \left(\mathbf{x}_{(k)}^{n+1} \right) \Delta \mathbf{x}_{(k)} - \mathbf{K}^p \left(\mathbf{x}_{(k)}^{n+1} \right) \Delta \mathbf{x}_{(k)} - \mathbf{C} \mathbf{v}_{(k+1)}^{n+1} \end{aligned} \quad (22)$$

With $\Delta \mathbf{x}_{(k)} = \Delta \mathbf{v}_{(k)} \Delta t$, we get

$$\begin{aligned} \mathbf{A} \Delta \mathbf{v}_{(k)} &= \mathbf{b} \\ \mathbf{A} &= \mathbf{M} + \Delta t \mathbf{C} + \Delta t^2 (\mathbf{K}^{elastic} \left(\mathbf{x}_{(k)}^{n+1} \right) + \mathbf{K}^p \left(\mathbf{x}_{(k)}^{n+1} \right)) \\ \mathbf{b} &= \mathbf{M} \left(\mathbf{v}^n - \mathbf{v}_{(k)}^{n+1} \right) + \Delta t \left(\mathbf{f}^{elastic} \left(\mathbf{x}_{(k)}^{n+1} \right) + \mathbf{f}^{ext} + \mathbf{f}^p \left(\mathbf{x}_{(k)}^{n+1} \right) - \mathbf{C} \mathbf{v}_{(k)}^{n+1} \right) \end{aligned} \quad (23)$$

$\Delta \mathbf{v}$ is solved by Conjugate Gradient method and $\mathbf{v}^{n+1} = \mathbf{v}^n + \Delta \mathbf{v}$ is the velocities in time $n + 1$. In conclusion, The detailed process in lines 5–9 of Algorithm 1 is

1. Compute the deformation gradient of each element using Eq. 4.
2. Compute discrete nodal elastic forces and pressure forces using Eqs. 8 and 15.
3. Construct system linear equation using Eq. 23.
4. Solve sequences of approximations $\mathbf{x}_{(k)}^{n+1} : \mathbf{x}_{(0)}^{n+1}, \mathbf{x}_{(1)}^{n+1}, \mathbf{x}_{(2)}^{n+1}, \dots$ and $\mathbf{v}_{(k)}^{n+1} : \mathbf{v}_{(0)}^{n+1}, \mathbf{v}_{(1)}^{n+1}, \mathbf{v}_{(2)}^{n+1}, \dots$ and compute the variations $\delta \mathbf{f}_i^{elastic}$, $\delta \mathbf{f}_i^p$ during the process using Eqs. 11 and 17.
5. $\mathbf{v}^{n+1} = \mathbf{v}_{(k+1)}^{n+1}$, $\mathbf{x}^{n+1} = \mathbf{x}_{(k+1)}^{n+1}$.

7 Volume Preservation

Volume preservation is an important feature of balloons. During deformation, the volume of gas is changing while the volume of balloon remains the same (see Fig. 3).

There are three vertices and one thickness property in an element. For the convenience of calculation, we convert the thickness property to a virtual vertex, which is located in the center of the upper triangle of three-prism (see Fig. 3). After volume preservation, we will convert the virtual vertex to thickness property.

The volume constraint is defined as the equality of initial volume and current volume of the balloon

$$C(\mathbf{x} + \Delta \mathbf{x}) = V^b(\mathbf{x} + \Delta \mathbf{x}) - V_0^b = 0 \quad (24)$$

where \mathbf{x} are current positions of $N + N_1$ vertices (N_1 are the virtual vertices). We need to solve the correction $\Delta \mathbf{x}$, which meet the constraint. Linearize C

$$C(\mathbf{x} + \Delta \mathbf{x}) \approx C(\mathbf{x}) + \frac{\partial C(\mathbf{x})}{\partial \mathbf{x}} \Delta \mathbf{x} \quad (25)$$

And compute the volume of the balloon from the volume of all three-prisms

$$V^b(\mathbf{x}) = \sum_e \frac{1}{2} \mathbf{e}_{14} \cdot (\mathbf{e}_{12} \times \mathbf{e}_{13}) \quad (26)$$

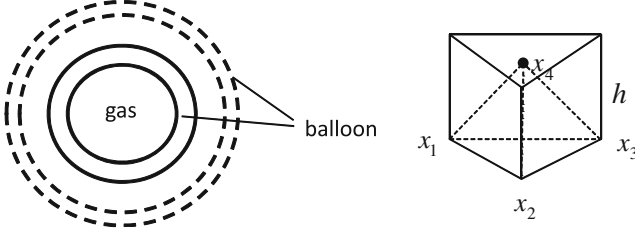


Fig. 3. Volume preservation of balloon. Left: The solid line represents the balloon before stretching, and the dot line represents the balloon after stretching, which is thinner than before. Right: The virtual vertex in volume preserving algorithm.

The correction is solved by position based method [28]

$$\Delta \mathbf{x}_i = - \frac{(N + N_1) w_i}{\sum_j w_j} \frac{w_i C(\mathbf{x})}{\sum_j \left\| \frac{\partial C(\mathbf{x})}{\partial \mathbf{x}_j} \right\|^2} \cdot \frac{\partial C(\mathbf{x})}{\partial \mathbf{x}_i} \quad (27)$$

where w_i are weights $w_i = 1/m_i'$. The new masses are computed using Eq. 19 with weight 1/6 for original three vertices and 1/2 for the virtual vertex. The derivation of C to vertex is easy to derive

$$\begin{aligned} \frac{\partial C(\mathbf{x})}{\partial \mathbf{x}_i} &= \sum_e \frac{\partial C^e(\mathbf{x})}{\partial \mathbf{x}_i} \\ &= \sum_{e: t_1^e=i} \frac{1}{2} \mathbf{e}_{24} \times \mathbf{e}_{23} + \sum_{e: t_2^e=i} \frac{1}{2} \mathbf{e}_{13} \times \mathbf{e}_{14} + \sum_{e: t_3^e=i} \frac{1}{2} \mathbf{e}_{14} \times \mathbf{e}_{12} + \sum_{e: t_4^e=i} \frac{1}{2} \mathbf{e}_{12} \times \mathbf{e}_{13} \end{aligned} \quad (28)$$

where $t_1^e, t_2^e, t_3^e, t_4^e$ are the four indices of the vertices belonging to element e .

The volume of balloon is preserved as follows

ALGORITHM 2. VOLUME PRESERVATION.

```

1  for all elements  $e$ 
2      do Compute the virtual vertex  $\mathbf{x}_4^e$  and add it to vertices matrix  $\mathbf{x}$ 
3          Compute new masses  $m_i'$  using Eq. 19
4          Using Eq. 28 to compute  $\frac{\partial C^e(\mathbf{x})}{\partial \mathbf{x}_i}$  and add it to  $\frac{\partial C(\mathbf{x})}{\partial \mathbf{x}_i}$ 
5  for all vertices  $i$ 
6      do Using Eq. 27 to compute  $\Delta \mathbf{x}_i$ 
7           $\mathbf{x}_i = \mathbf{x}_i + \Delta \mathbf{x}_i$ 
8  for all elements  $e$ 
9      do Compute thickness  $h^e$  from virtual vertices

```

8 Results

All of our examples are tested on an Intel Core i7 3.4 Ghz CPU with 16 GB memory. In order to explore the capabilities of our method, we make tests with

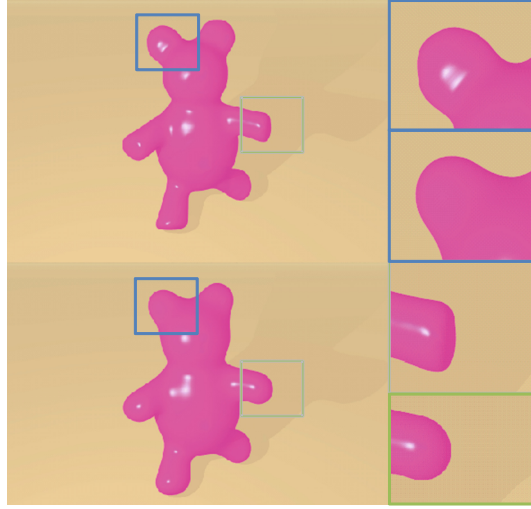


Fig. 4. Inflating result of the teddy. Left top is the undeformed shape, left bottom is the deformed shape and right are detail images.

a variety of different shapes from The Princeton Shape Benchmark as the undeformed balloons and the results are rendered by POV-Ray. For each shape, an inflating animation is made and we choose some frames to analysis.

Figure 4 shows the inflating result of teddy. The shape of teddy’s head is some of a big sphere, on which the ear is some of a small hemisphere. During the animation, the big parts such as the head and the body are inflated fast, while the small ones are inflated slowly like the ears. The reason is that the big parts have equable curvature while the small ones have steep curvature. The steeper the nodal’s curvature is, the smaller pressure force it gets (see Fig. 5). Additionally, as the head is inflated faster than the ears, the vertices connecting the two parts will have larger elastic forces than those belong to ears and so the base area of the ears will enlarged while the height is shorten correspondingly. As a result, the ears seem like shrinking and merged into the head just like the balloons in reality do.

More results are given in Fig. 6. From the results, we can see that the big parts are inflated faster than the small ones and they are merged gradually. However, parts with the similar size will keep their shapes and be inflated separately. Additionally, the regions between different parts become flat gradually. The rough regions in the Vase model get smooth while the plates in the Cube & Cylinder model get curved. The balloons simulated by our algorithm act just like the real balloons.

Thickness Analysis and Comparison. The balloons in same shape are deformed in different ways with different thicknesses. With larger thickness, the balloons are more difficult to stretch due to larger elastic forces. We experiment with different thicknesses of 0.3 mm, 0.2 mm and 0.1 mm in Fig. 7.

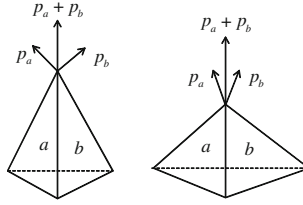


Fig. 5. The relationship with curvature and pressure force.



Fig. 6. Results of more shapes. Left column are undeformed shapes and right column are deformed shapes.

From the results, we can see that the balloon is deformed more slowly with larger thickness but the effect of emerging is more obvious. With thickness 0.3mm, the ears are barely inflated and the deformation is mainly from the stretch of the connecting part with the head. At last, the two parts get merged quickly. With thickness 0.1 mm, the deformation of the ears is mainly from the inflating of itself and the ears never be merged with the head. With thickness 0.2mm, the deformation of the ears is the combination of the two effects.

The results show that thickness is an important property of balloons. Different thicknesses will cause different effects, which is ignored in other methods. The last figure in Fig. 7 is the result of Skouras et al. [19]. In their method, the deformation gradient is complex and the volume is preserved by brutally setting

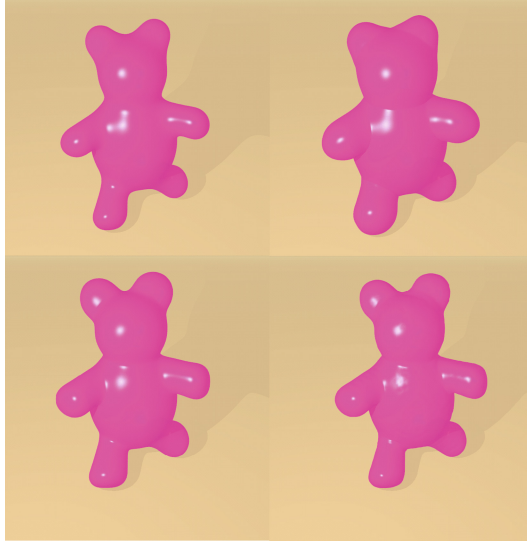


Fig. 7. Results of different thicknesses and comparison with the result of Skouras et al. [19]. Left top is the result of thickness 0.3, right top is the result of thickness 0.2, left bottom is the result of thickness 0.1, right bottom is the result of Skouras et al. [19].

the thickness to balance the changing of the area of triangles. Thus the method can not realize the different effects of different thicknesses.

Performance. We provide computation times and volume ratios for all examples shown in this paper in Table 1. The volume ratio is the ratio of current volume and the initial volume of the balloon. The thickness of all models is 0.3 mm and the number of iterations of Newton-Raphson method is 3. We calculate the average computation time and volume ratio of 300 frames.

It can be seen that the computation times are acceptable and the volume ratios of all models are almost 1, which means that our volume preservation algorithm works well. Due to the rough surface, there are too much details in the Vase model, thus the volume preservation performance is not as good as other models.

Table 1. Computation times and volume ratio.

| Model | #vertices | #faces | Time [ms] | Volume ratio |
|-----------------|-----------|--------|-----------|--------------|
| Teddy | 11090 | 22176 | 278.0 | 1.0000026 |
| Bird | 7849 | 15694 | 187.2 | 1.0000028 |
| Holes | 5884 | 11776 | 207.6 | 1.0000065 |
| Vase | 14599 | 29194 | 522.6 | 0.9924975 |
| Cube & Cylinder | 15505 | 31006 | 484.2 | 1.0000036 |

9 Conclusion

In this paper, we present a simple thin shell deformation method, based on finite element method. We focus on the objects with large stretch deformation and little bend deformation, balloons are typical objects of which. Considering the special features of balloons, we introduce gas model and volume preservation algorithm to simulate the inflating process of balloons. We experiment with a variety of different shapes from The Princeton Shape Benchmark and get good results. Compared to other methods, we can realize the effects of different thicknesses.

Although we can simulate balloons with different thicknesses, the thickness of all elements should be the same when initialized. However, the real balloons often have different thicknesses in different parts, which will be explored in our future work. Additionally, adding external forces on balloons such as grabbing or pinching is also an interesting topic.

Acknowledgments. The models are provided by The Princeton Shape Benchmark.

References

1. Bonet, J., Wood, R.D., Mahaney, J., et al.: Finite element analysis of air supported membrane structures. *Comput. Methods Appl. Mech. Eng.* **190**(5–7), 579–595 (2000)
2. Platt, J., Fleischer, K., Terzopoulos, D., et al.: Elastically deformable models. *Comput. Graph.* **21**(4), 205–214 (1987)
3. Provot, X.: Deformation constraints in a mass-spring model to describe rigid cloth behavior. In: *Graphics Interface*, pp. 147–154 (1995)
4. Desbrun, M., Schröder, P., Barr, A.: Interactive animation of structured deformable objects. *Graph. Interface* **99**(5), 10 (1999)
5. James, D.L., Pai, D.K.: ArtDefo: accurate real time deformable objects. In: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press/Addison-Wesley Publishing Co., pp. 65–72 (1999)
6. O’Brien, J.F., Hodgins, J.K.: Graphical modeling and animation of brittle fracture. In: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press/Addison-Wesley Publishing Co., pp. 137–146 (1999)
7. Gibson, S.F.F., Mirtich, B.: A survey of deformable modeling in computer graphics. Technical report, Mitsubishi Electric Research Laboratories (1997)
8. Nealen, A., Müller, M., Keiser, R., et al.: Physically based deformable models in computer graphics. *Comput. Graph. Forum* **25**(4), 809–836 (2006). Blackwell Publishing Ltd.,
9. Eischen, J.W., Deng, S., Clapp, T.G.: Finite-element modeling and control of flexible fabric parts. *IEEE Comput. Graph. Appl.* **16**(5), 71–80 (1996)
10. Janíski, L., Ulbricht, V.: Numerical simulation of mechanical behaviour of textile surfaces. *ZAMM J. Appl. Math. Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik* **80**(S2), 525–526 (2000)

11. Müller, M., Dorsey, J., McMillan, L., et al.: Stable real-time deformations. In: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 49–54. ACM (2002)
12. Eitzmuß, O., Keckeisen, M., Straßer, W.: A fast finite element solution for cloth modelling. In: Proceedings of the 11th Pacific Conference on Computer Graphics and Applications, 2003, pp. 244–251. IEEE (2003)
13. McAdams, A., Zhu, Y., Selle, A., et al.: Efficient elasticity for character skinning with contact and collisions. *ACM Trans. Graph. (TOG)* **30**(4), 37 (2011). ACM
14. Arnold, D.: Questions on shell theory. In: Workshop on Elastic Shells: Modeling, Analysis, and Computation (2000)
15. Cirak, F., Scott, M.J., Antonsson, E.K., et al.: Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision. *Comput. Aided Des.* **34**(2), 137–148 (2002)
16. Green, S., Turkiyyah, G., Storti, D.: Subdivision-based multilevel methods for large scale engineering simulation of thin shells. In: Proceedings of the Seventh ACM Symposium on Solid Modeling and Applications, pp. 265–272. ACM (2002)
17. Grinspun, E., Krysl, P., Schröder, P.: CHARMS: a simple framework for adaptive simulation. *ACM Trans. Graph. (TOG)* **21**(3), 281–290 (2002). ACM
18. Grinspun, E., Hirani, A.N., Desbrun, M., et al.: Discrete shells. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 62–67. Eurographics Association (2003)
19. Skouras, M., Thomaszewski, B., Bickel, B., et al.: Computational design of rubber balloons. *Comput. Graph. Forum* **31**(2pt4), 835–844 (2012). Blackwell Publishing Ltd.,
20. Guendelman, E., Selle, A., Losasso, F., et al.: Coupling water and smoke to thin deformable and rigid shells. *ACM Trans. Graph. (TOG)* **24**(3), 973–981 (2005)
21. Robinson-Mosher, A., Shinar, T., Gretarsson, J., et al.: Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Trans. Graph. (TOG)* **27**(3), 46 (2008). ACM
22. Batty, C., Bertails, F., Bridson, R.: A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph. (TOG)* **26**(3), 100 (2007)
23. Chen, Z., Feng, R., Wang, H.: Modeling friction and air effects between cloth and deformable bodies. *ACM Trans. Graph.* **32**(4), 96–96 (2013)
24. Müller, M., Keiser, R., Nealen, A., et al.: Point based animation of elastic, plastic and melting objects. In: Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer animation, pp. 141–151. Eurographics Association (2004)
25. Bargteil, A.W., Wojtan, C., Hodgins, J.K., et al.: A finite element method for animating large viscoplastic flow. *ACM Trans. Graph. (TOG)* **26**(3), 16 (2007)
26. Irving, G., Schroeder, C., Fedkiw, R.: Volume conserving finite element simulations of deformable models. *ACM Trans. Graph.* **26**(3) (2007)
27. Hong, M., Jung, S., Choi, M.H., et al.: Fast volume preservation for a mass-spring system. *IEEE Comput. Graph. Appl.* **26**(5), 83–91 (2006)
28. Müller, M., Heidelberger, B., Hennix, M., et al.: Position based dynamics. *J. Vis. Commun. Image Representation* **18**(2), 109–118 (2007)
29. Müller, M.: Hierarchical position based dynamics (2008)
30. Diziol, R., Bender, J., Bayer, D.: Robust real-time deformation of incompressible surface meshes. In: Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 237–246. ACM (2011)
31. Baraff, D., Witkin, A.: Large steps in cloth simulation. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, pp. 43–54. ACM (1998)